Face recognition security system on public cloud infrastructures

Ramón Pintado Martínez

MASTER EN INVESTIGACIÓN EN INFORMÁTICA, FACULTAD DE INFORMÁTICA, UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería de Computadores

Madrid, 20 de junio de 2014

Director: Jose Luis Vazquez-Poletti

Calificación: Sobresaliente

Autorización de difusión

RAMÓN PINTADO MARTÍNEZ

Madrid, 20 de junio de 2014

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “Face recognition security sistem on public cloud”, realizado durante el curso académico 2013-2014 bajo la dirección de Jose Luis Vazquez-Poletti en el Departamento de Arquitectura de Computadores, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Dedicatoria

A B.G.D., mis padres y amigos por su apoyo incondicional

Agradecimientos

Quiero expresar mi más sincera gratitud a:

- mi familia, por enseñarme que la vida es un reto constante y que nunca debo bajar los brazos.

- mi director de proyecto, por su incalculable ayuda e interés en mi trabajo.

Tabla de contenido

Índice de Figuras

Índice de tablas

Resumen

Hoy en día la seguridad es un tema clave en nuestra sociedad. Lugares públicos con gran afluencia de gente como aeropuertos, estaciones de tren o incluso eventos sociales multitudinarios cuentan con sistemas de seguridad con video vigilancia. Este proyecto propone dos modelos para la implantación de un sistema de seguridad utilizando la tecnología emergente del reconocimiento facial y dotado de suficiente potencia para hacer frente a grandes cantidades de datos gracias a la tecnología del cloud público. El primer modelo, preparado para escenarios estáticos como garitas de seguridad, cajeros bancarios o pasillos unidireccionales, proporciona resultados muy precisos gracias al control de la iluminación, pose o expresión de los individuos analizados. El segundo modelo está preparado para escenarios dinámicos como cámaras de vigilancia IP en los cuales las variables del entorno no son controladas, actuando como un excelente sistema de apoyo del primer modelo. Estos modelos permiten la identificación de personas cuyas fotografías están incluidas en las grandes bases de datos de los diferentes organismos de seguridad en el mundo. Además, gracias a la gran variedad de máquinas que nos ofrece el cloud público, se presenta un modelo basado en árbol de decisión, en el que el usuario puede determinar de forma sencilla la cantidad de máquinas que necesita dependiendo de flujo de personas, el tiempo máximo para la identificación y el coste monetario que está dispuesto a asumir; un rendimiento óptimo o máxima potencia.

Palabras clave:

Reconocimiento facial, video vigilancia, computación en la nube, seguridad.

Abstract

Nowadays security is a key issue in our society. Public places with large numbers of people such as airports, train stations or even crowded social events have security systems with video surveillance. This project proposes two models for the implementation of a security system using the emerging technology of facial recognition and equipped with enough power to handle large amounts of data using public cloud technology. The first model, ready for static scenarios like security booths, cash machines or one-way corridors, provides very accurate results by controlling lighting, pose or expression of individuals analysed. The second model is prepared for dynamic environments, such as IP surveillance cameras, in which environment variables are not controlled, acting as an excellent support system of the first model. These models allow the identification of persons whose photographs are included in the large databases of different security agencies in the world. Additionally, thanks to the wide variety of machines offered by public cloud, based on decision tree model is presented, in which the user can easily determine the number of machines he needs depending on the flow of people, maximum time for the identification and monetary cost he is willing to take; the best performance or maximum power.

Key words:

Face recognition, surveillance, cloud computing, security.

PARTE I

Motivación y objetivos

Capítulo 1

1. Introducción

El crecimiento del *cloud computing* y del almacenamiento *cloud* ha sido el precursor y facilitador de la aparición de la *big data*. El uso del *cloud* tiene ventajas significativas frente a los despliegues físicos tradicionales. Sin embargo, la gran cantidad de plataformas *cloud* que existen a veces tienen que ser integradas con arquitecturas tradicionales. Esto conduce a un dilema a los encargados de tomar decisiones en los grandes proyectos. Normalmente, estos proyectos son impredecibles, presentan estallidos tanto de computación como de almacenamiento. Buscar la opción más óptima en cuanto a rendimiento y el coste de esta nueva infraestructura en la nube no es para nada sencillo pero es primordial para que los proyectos salgan adelante.

Hace una década cualquier proyecto IT que necesitara recursos informáticos fiables conectados a internet tenía que alquilar o colocar hardware físico en uno o varios centros de datos. Hoy en día, cualquiera puede alquilar tiempo de computación y almacenamiento de cualquier tamaño. La mayoría de servicios *cloud* son del tipo *pay-as-you-go*, que quiere decir, que por unos pocos dólares podemos tener acceso a máquinas virtuales para computación y por unos cuantos cientos de dólares más, cualquiera puede tener acceso por unas horas a la potencia de un supercomputador. El *cloud computing* emplea la visualización de los recursos de computación para ejecutar numerosos servidores virtuales en la misma máquina. De esta manera los proveedores logran economías de escala que permiten precios bajos y una facturación basada en intervalos pequeños de tiempo, por ejemplo, de una hora. Esta estandarización permite una alta elasticidad y una gran disponibilidad para el cálculo de necesidades. La disponibilidad no se obtiene por el gasto de los recursos para

garantizar la fiabilidad en un solo caso, sino por su capacidad de intercambio y una infinidad de reemplazos.

Las consecuencias para un proyecto IT o empresa que utiliza la computación en la nube son significativos y cambia el enfoque tradicional de la planificación y la utilización de los recursos. En primer lugar, la planificación de los recursos se vuelve menos importante. Se requiere para calcular el coste de los escenarios para establecer la viabilidad del proyecto o producto. Sin embargo, la implementación y la eliminación de los recursos automáticamente en función de la demanda tienen que ser bien enfocados para tener éxito. La escala vertical y horizontal se hace viable una vez que se convierte en un recurso fácil de implementar.

La escala horizontal se refiere a capacidad de reemplazar una máquina pequeña por una más grande para hacer frente al aumento de la demanda. Esto se consigue preparando varios recursos disponibles para poder cambiar entre ellos. La demanda es difícil de prever a pesar de los esfuerzos de planificación. La escala vertical logra elasticidad añadiendo instancias adicionales las cuales, cada una de ellas, se encarga de una parte de la demanda. Software como Hadoop están específicamente diseñados para sistemas distribuidos y aprovechar las ventajas de la escala vertical. En cuanto a los proveedores de *cloud*, Amazon se postula como el mayor proveedor de *cloud* público con el llamado Amazon Web Service (AWS), el cual cuenta con multitud de herramientas para gestionar el entorno *cloud*.

Los sistemas de seguridad que utilizan reconocimiento facial son cada vez más utilizados. Grandes empresas, casinos, edificios públicos o gubernamentales y zonas con el perímetro restringido dónde existe un conjunto de personas conocidas que trabajan en esos lugares son los perfectos escenarios para el uso de estos sistemas. Las nuevas técnicas de reconocimiento, como el reconocimiento en 3D o el análisis de la piel, junto a los algoritmos clásicos dotan a estos sistemas de mayor precisión y rapidez.

Las aplicaciones son muy diversas pero siempre manteniendo un factor común. El conjunto de fotos con las que se compara un individuo es pequeño o limitado, es decir, el número de empleados, conocidos contadores de cartas en casinos, funcionarios o soldados en una base militar. Siempre que esta base de datos sea pequeña estos sistemas funcionan a tiempo real prácticamente. En el caso contrario, las aplicaciones comienzan a producir unos tiempos de espera incontrolables e inaceptables en sistemas de seguridad. Es en este punto donde podemos aprovechar la potencia del *cloud* público para aumentar las posibilidades de estos sistemas de seguridad. Aumentando la potencia de computación de forma elástica según sea necesitada y por un bajo precio podemos ampliar los casos de uso de estos sistemas. Por ejemplo, una terminal de un aeropuerto por la cual pasan cada día cientos de miles de personas necesita grandes medidas de seguridad. Podemos utilizar estos sistemas de reconocimiento facial para buscar entre toda la multitud de gente que embarca y desembarca de los aviones a los criminales buscados por todo el FBI o la Interpol. Bases de datos de 10000, 15000 o 20000 personas con las cuales se puede trabajar casi a tiempo real dando una gran potencia a estos sistemas.

Dada la cantidad de instancias necesarias para hacer funcionar estos sistemas y el dinero que esto supone, es necesario un estudio previo de los escenarios. Obtener la forma más óptima de manejar estas instancias es clave para que estos sistemas sean viables tanto computacionalmente como por su gasto. La gran cantidad de máquinas que nos ofrecen los proveedores de cloud y sus diferentes opciones de pago nos permiten adaptar nuestro sistema a las necesidades que mejor nos convengan: una gran potencia de computación en ciertos momentos para una seguridad exhaustiva o quizás un rendimiento más moderado pero a un coste más bajo.

La cantidad de información que se maneja en estos sistemas sería enorme por lo que se convierte en un problema de big data con una gran necesidad de potencia de computación.

Capítulo 2

2. Trabajo Previo

En el siguiente capítulo es dada una visión general sobre los sistemas de seguridad que utilizan reconocimiento facial. En primer lugar, se presenta un resumen de los distintos métodos de reconocimiento facial que existen, destacando las ventajas y desventajas de cada uno. Después, se da a conocer el estado del arte centrándonos en los sistemas de seguridad basados en reconocimiento facial que ya han sido utilizados en diversos escenarios, comparando sus características y a la cantidad de información a la que eran capaces de enfrentarse. Más tarde, se analizarán las distintas formas de gestionar la gran cantidad de información en forma de imágenes, conocida como *big data*, para atender a los conceptos de High Performance Computing y High Throughput Computing.

Finalmente, se introducen diferentes capas *cloud*, servicios y principales modelos básicos, explicando todos los beneficios sobre la migración al *cloud* y sus posibles inconvenientes.

2.1 Reconocimiento facial

2.1.1 Ventajas del reconocimiento facial

Las técnicas biométricas se postulan como la opción más prometedora para el reconocimiento de personas en los últimos años, ya que, en lugar de realizar la autenticación de las personas para acceder a sus dominios físicos o virtuales a través de contraseñas, PINs, tarjetas inteligentes o de plástico, fichas y así sucesivamente, éstos métodos examinan las características fisiológicas y/o de comportamiento de un individuo con el fin de determinar su identidad. Las contraseñas y PINs son difíciles de recordar ya hoy en día es muy común tener una gran cantidad de ellos. Además, pueden ser robados, duplicados o adivinados, o simplemente dañados como en el caso de las tarjetas magnéticas. Sin embargo, los rasgos biológicos de una persona no pueden estar fuera de lugar, caer en el olvido, ser robados o falsificados.

Dentro de las tecnologías biométricas se incluye la identificación de individuos a través de características fisiológicas como: la cara, huellas dactilares, geometría de la mano o los dedos, venas de las manos, palmas de las manos, el iris, la retina o incluso las orejas o la voz [1]. El reconocimiento facial ofrece grandes ventajas frente al resto de métodos biométricos. La mayoría de estas tecnologías requieren la voluntariedad del usuario. Por ejemplo, el usuario necesitar poner su mano sobre un lentor de huellas o su ojo delante de una cámara para un análisis de retina. Sin embargo, el reconocimiento facial puede ser realizado de forma pasiva sin ninguna interacción del usuario a parte de las imágenes obtenidas a distancia a través de una cámara. Esto es beneficioso para usar estos sistemas con fines de seguridad y vigilancia.

Además, existen problemas con la adquisición de información en los otros métodos biométricos. Técnicas como leer las huellas dactilares pueden ser poco efectivas o inútiles si las huellas o la piel están dañadas a causa de una quemadura

o algún tipo de lesión. Los lectores de retina o iris requieren equipos muy caros y los reconocimientos de voz pueden ser susceptibles a ruido externo en sitios públicos o grabaciones por teléfono móvil.

Sin embargo, imágenes faciales de gran calidad pueden ser tomadas fácilmente con un par de cámaras con un coste muy reducido. Los algoritmos de reconocimiento facial pueden compensar el ruido de las fotografías, la orientación, la iluminación y la escala. Por último, desde el punto de vista de la salud e higiénico, algunos de estos métodos pueden exponer a los usuarios a los gérmenes o impurezas del resto de usuarios. En cambio, el reconocimiento facial no presenta ningún riesgo para la salud.

2.1.2 Dificultades generales

El reconocimiento facial es un caso complicado del reconocimiento de objetos. La dificultad de este problema viene del hecho de que en su forma más común, es decir, las imágenes de caras en vista frontal parecen ser más o menos iguales y las diferencias entre ellas son bastante sutiles. En consecuencia, las imágenes de caras frontales forman un grupo muy denso en el espacio de imagen que hace que sea prácticamente imposible para las técnicas de reconocimiento con patrones tradicionales discriminar con precisión entre todas ellas con un alto grado de éxito [2].

Hay que tener en cuenta que la cara humana no es un objeto único ni rígido. De hecho, numerosos factores pueden causar variaciones en la apariencia de la cara. Las fuentes de variación de los rasgos faciales pueden ser clasificadas en dos grupos [3]:

- Factores Intrínsecos: Se deben única y exclusivamente a la naturaleza física de la cara y son independientes del observador. Se dividen en dos clases:

    - Intrapersonales: Son responsables de la variación de la apariencia facial de una misma persona siendo algunos ejemplos la edad, la expresión facial y la parafernalia facial (barba, gafas, cosméticos etc).
    - Interpersonales: Son responsables de las diferencias en la apariencia facial de distintas personas, siendo algunos ejemplos la etnia y el género.

- Factores Extrínsecos: Causan la variación de los rasgos faciales a través de la interacción de la luz con la cara y el observador. Se incluyen la iluminación, pose, parámetros de escala y de imagen (resolución, enfoque, ruido etc).

No obstante, la mayoría de sistemas de reconocimiento facial funcionan bien bajo diversas condiciones pero su rendimiento se degrada rápidamente cuando trabajan bajo condiciones que no son reguladas.

2.1.3 Métodos de reconocimiento facial

El método para adquirir imágenes faciales depende del tipo de aplicación. Por ejemplo, las aplicaciones de vigilancia funcionan mejor capturando imágenes faciales por medio de una cámara de vídeo mientras que las investigaciones de bases de datos de imágenes pueden requerir imágenes de intensidad estática tomadas por una cámara estándar. Otras aplicaciones, tales como accesos a dominios de seguridad superiores pueden requerir incluso la renuncia de la no intrusividad del individuo exigiendo al usuario estar delante de un escáner 3D o un sensor de infrarrojos.

Por lo tanto, dependiendo de la metodología de adquisición de datos faciales, las técnicas de reconocimiento facial se pueden dividir en tres categorías: los métodos que operan sobre imágenes de intensidad, los que se ocupan de las secuencias de video y los que requieren otros datos sensoriales, como la información 3D o infrarrojos.

2.1.3.1 Imágenes de Intensidad

Los métodos de reconocimiento facial a través de imágenes de intensidad pueden ser clasificados en dos categorías: *featured-based* y holística [4] [5] [6].

2.1.3.1.1 Featured-Based:

Los enfoques basados en *featured-based* primero procesan la imagen de entrada para identificar, extraer y medir las características faciales como los ojos, la boca, la nariz, etc, así como otras marcas de referencia para después calcular las relaciones geométricas entre esos puntos faciales, reduciendo de este modo la imagen de entrada a un vector de características geométricas. Posteriormente se emplean técnicas de reconocimiento con patrones estadísticos estándar para emparejar las caras.

La mayoría de los primeros trabajos en reconocimiento facial automático utilizaban esta técnica. Uno de los primeros intentos fue por parte de Kanade [7] que empleo métodos de procesado de imágenes simples para extraer un vector de 16 rasgos faciales. Utilizando esas distancias y la distancia Euclídea obtuvo un pico de rendimiento del 75% en una base de datos de 20 personas usando 2 imágenes por persona.

Más reciente, Cox [8] reporto un rendimiento de reconocimiento de un 95% en una base de datos de 685 imágenes (una imagen para cada individuo) usando un vector de 30 dimensiones derivado de 35 puntos faciales (Fig. 2.1). Sin embargo, los puntos faciales eran extraídos manualmente por lo que el rendimiento se degradaría si fuera automático. En general los algoritmos de reconocimiento automático tienden a degradar el rendimiento y requieren una potencia computacional considerable.

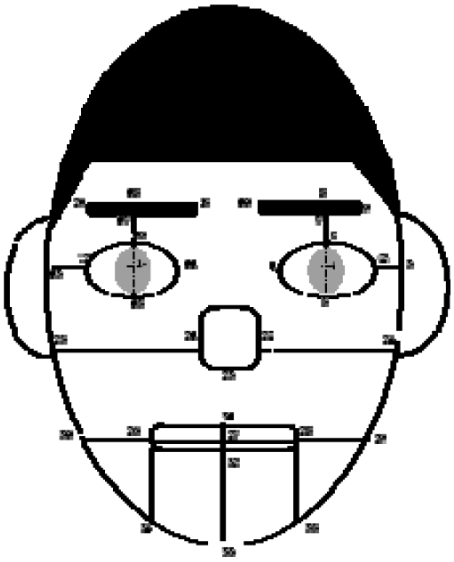

Fig. 2.1 Detección manual de 35 rasgos faciales [8]

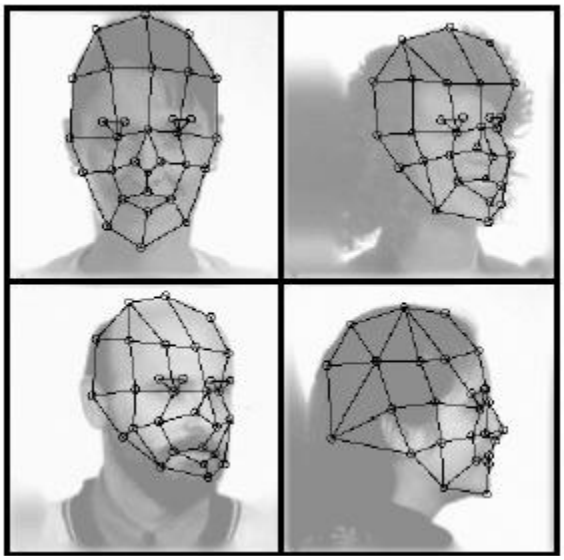

Fig. 2.2 Reconocimiento facial con red

Otro enfoque conocido de la técnica *feature-based* es el método de comparación a través de grafos elásticos propuesto por Wiskott [9]. Esta técnica está basada en estructuras dinámicas enlazadas. El grafo de un individuo se genera de la siguiente forma: un conjunto de puntos en la cara son elegidos. Cada punto es un nodo de un grafo conexo completo y se etiqueta con las respuestas de los filtros de *Gabor* aplicando una ventana alrededor del punto. Un conjunto representativo de estos grafos son combinados dentro de una estructura llamada *face bunch graph*. Una vez el sistema tiene esta estructura puede generar grafos para nuevas caras de forma automática. Usando esta arquitectura el ratio de reconocimiento alcanza el 98% usando una base de datos de 250 individuos. El sistema ha sido mejorado para permitir tratar con diferentes poses (Fig. 2.2), manteniendo el mismo rendimiento en cada una [9].

2.1.3.1.1.1 Ventajas y desventajas featured-based

La principal ventaja que ofrecen las técnicas *featured-based* es que, dada que la extracción de los puntos faciales precede al análisis realizado para comparar los rasgos faciales con un individuo conocido, tales métodos son relativamente robustos

frente a las variaciones de posición en las imágenes de entrada [3]. En principio los esquemas no presentan cambios frente a variaciones de tamaño, orientación o luz [8]. Otra de sus ventajas es que presentan una alta capacidad de compactación y alta velocidad de comparación.

La principal desventaja de este método es la dificultad de automatizar la detección de los rasgos faciales y el hecho de que el responsable de implementar estos sistemas tienen que tomar la decisión arbitraria de cuáles son los puntos realmente importantes [10]. Después de todo, si el conjunto de características faciales carece de capacidad de discriminación, ninguna cantidad de computación posterior puede compensar esa deficiencia [8].

2.1.3.1.2 Holística:

El enfoque holístico trata de identificar caras usando representaciones globales, es decir, en vez de utilizar rasgos locales (puntos), utilizar descripciones basadas en la imagen completa. Este enfoque puede ser subdivido en dos grupos: estadísticos y aproximaciones IA.

2.1.3.1.2.1 Estadístico

En la versión simple de la aproximación holística, la imagen es representada como un array 2D de valores de intensidad. El reconocimiento es realizado directamente a través de la comparación entre la cara entrante y las otras caras almacenadas en la base de datos.

Aunque este enfoque ha sido demostrado [11] bajo circunstancias limitadas (es decir, igualdad de iluminación, escala, actitud, etc), es computacionalmente muy caro y sufre los inconvenientes habituales de los enfoques basados en correlaciones directas, tales como la sensibilidad frente a la orientación, tamaño, condiciones de iluminación variable (Fig. 2.3), desorden de fondo o ruido [12].

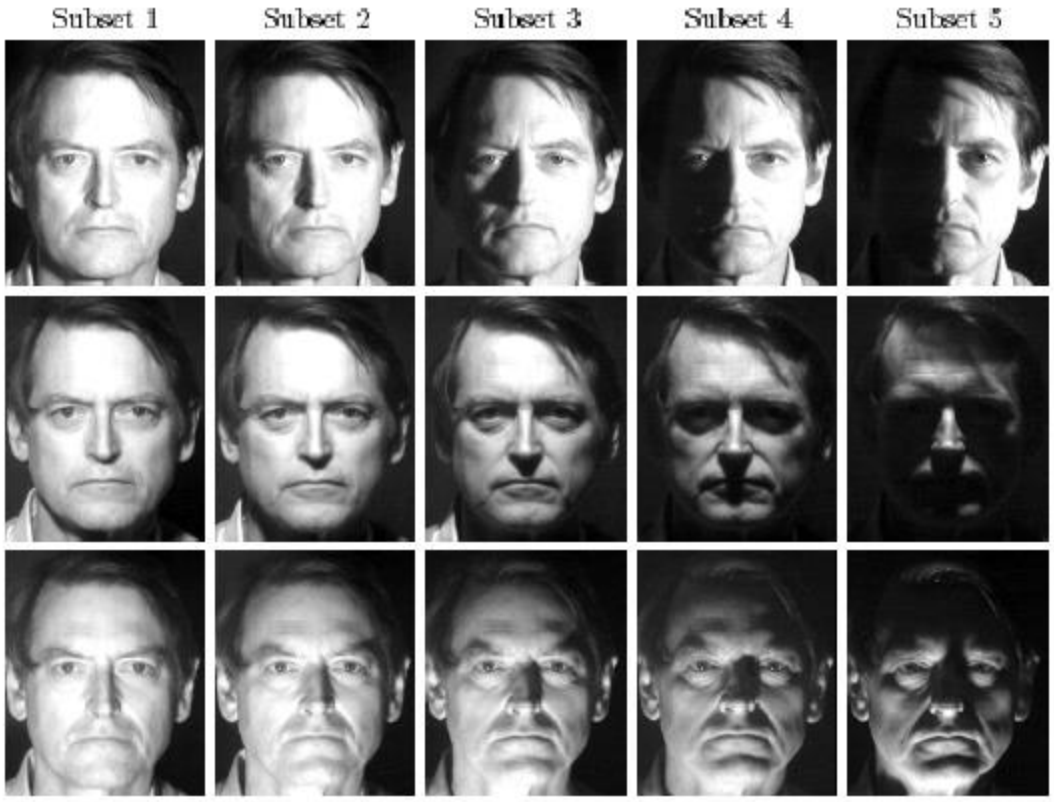

Fig. 2.3 Misma persona bajo condiciones variables de luminosidad parece completamente diferente

El mayor obstáculo para el rendimiento de los métodos de comparación directa es que tratan de realizar una clasificación en un espacio de muy alta dimensionalidad [12]. Para contener el problema de la dimensionalidad, se han propuesto numerosos esquemas que emplean métodos estadísticos de reducción de dimensionalidad para obtener y conservar las dimensiones de las características más relevantes antes de realizar el reconocimiento. Algunos de estos métodos son:

- Sirovich y Kirby [13] utilizaron del Análisis de Componentes Principales (PCA) [14] [15] para representar económicamente imágenes de rostros. Cualquier cara particular puede ser eficientemente representada utilizando el espacio de coordenadas *eigenpictures* y puede ser reconstruida utilizando un número pequeño de *eigenpictures* y las correspondientes proyecciones a lo largo de cada uno.

- Turk and Penland descubrieron, utilizando los datos de Sirovich y Kirby, que las proyecciones a lo largo de los eigenpictures podían ser usados para clasificar los rasgos para reconocer caras obteniendo ratios de reconocimiento del 96%, 85% y 64% dependiendo de la iluminación.

- Análisis de componentes independientes (ICA) [16], una generalización de PCA, trata de encontrar una descomposición y representación independiente

- Utilización del Análisis lineal discriminante de Fisher [17] que maximiza la relación de dispersión entre clase y la dispersión dentro de la clase. Es por tanto supuestamente mejor para la clasificación que PCA.

  Un estudio y comparación de estas técnicas para representar el subespacio para el reconocimiento facial es presentado en [18] y a algunos avances pueden ser encontrados en [19]

2.1.3.1.2.2 IA:

El enfoque de la IA utiliza herramientas como redes neuronales y técnicas de aprendizaje de máquinas para reconocer caras. Algunos ejemplos de métodos que pertenecientes a esta categoría son:

- Redes neuronales híbridas que combinaban imágenes locales de muestreo, es decir, un mapa de auto-organización neuronal que reducía la dimensionalidad y una red neuronal de convolución que protegía frente a variaciones de rotación, escala y deformaciones [20].

- Reemplazando el mapa de auto organización por la transformación de Karhunen-Loeve y la red de convolución por un perceptrón multicapa se

obtuvieron porcentajes de reconocimiento del 94.7% y 60% respectivamente (Fig. 2.4).

- Enfoque uno contra uno en los que se descompone el problema multiclase del reconocimiento facial en un número de problemas de clasificación binaria [21]. Para los clasificadores binarios con salidas probabilísticas se utilizaban acoplamiento de pares (*pair-wise couple* o PWC).

- Otros enfoques IA incluyen reconocimiento facial a través de *evolutionary pursuit* [22] [23] y técnicas de *boosting* [24], los cuales han resultado prometedores en escenarios complicados.

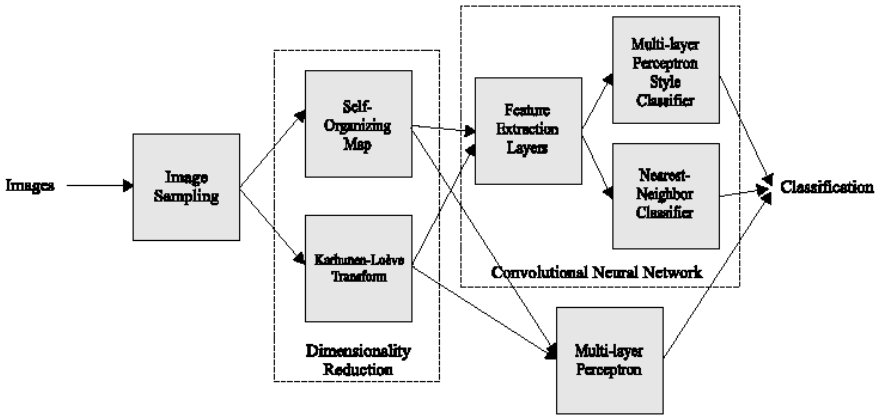

Fig. 2.4 Diagrama de alto nivel de un sistema de reconocimiento facial

2.1.3.1.2.3 Ventajas y desventajas de la holística:

La principal ventaja del enfoque holístico es que no destruyen información de las imágenes por centrarse únicamente en regiones o puntos de interés. Sin embargo, esta propiedad es un mayor defecto ya que asume que todos los pixels de la imagen son igual de importantes. En consecuencia, estas técnicas no sólo son computacionalmente costosas, además requieren un alto grado de correlación entre el test y las imágenes de entrenamiento. Sin embargo, existen numerosos algoritmos que mejoran o compensan las variaciones y la gran dimensionalidad produciendo resultados de reconocimiento que la técnica featured-based en general.

2.1.3.2. Vídeo secuencias:

Una de las principales aplicaciones del reconocimiento facial es la vigilancia con fines de seguridad, lo que implica reconocimiento facial a tiempo real a partir de una secuencia de imágenes tomadas por una cámara de vídeo. Una cantidad significativa de investigaciones se han dirigido a esta área en los últimos años. Los sistemas de reconocimiento facial a través de vídeo secuencias constan típicamente de tres módulos: uno para detectar la cara, otro para el rastreo o *tracking* y otro para el reconocimiento [25]. La mayoría de estos sistemas usan unos cuantos *frames* buenos y aplican alguna de las técnicas de reconocimiento para imágenes de intensidad para poder realizar las identificaciones [26].

Howell and Buxton [27] emplearon una red RBF de dos capas para el entrenamiento y usaron para la representación de los rasgos un filtro de diferencia gaussiana (DoG). El entrenamiento y el testeo fue hecho usando dos tipos de secuencias de imágenes: ocho secuencias tomadas en un entorno relativamente limitado y otras ocho secuencias tomadas en otro entorno mucho menos limitado. Estas secuencias van desde 62 a 94 frames de los cuales se eligen los mejores (Fig. 2.5 y Fig. 2.6).La precisión obtenida varía un poco. Un 99% usando 278 imágenes

para el entrenamiento y 276 para el test y un 67% usando 16 imágenes para el entrenamiento y 538 para el test.

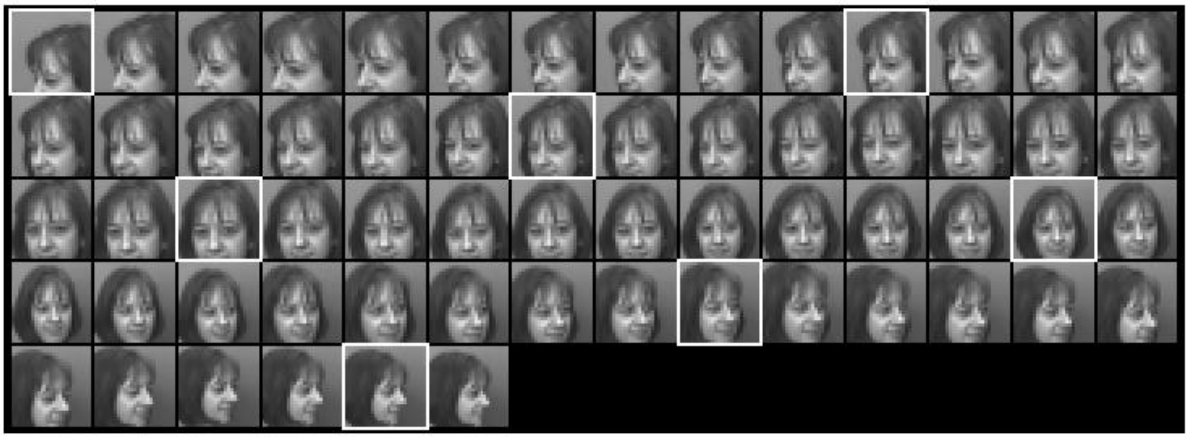

Fig. 2.5 Primeras ocho secuencias tomadas en un entorno limitado

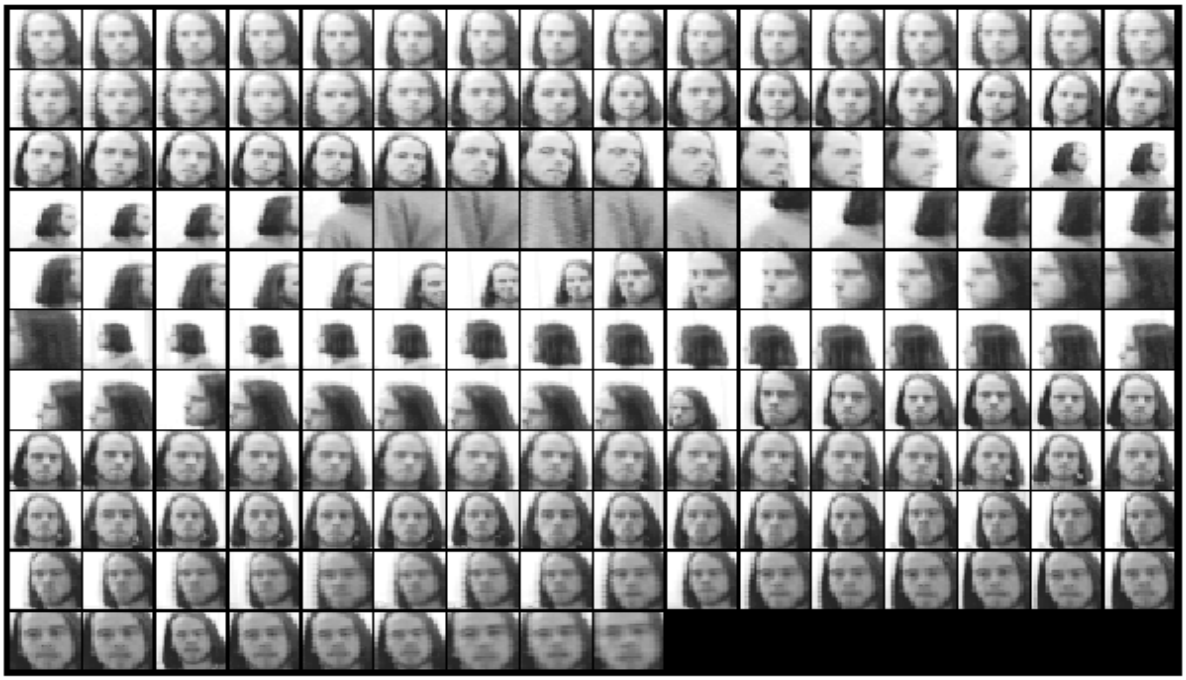

Fig. 2.6 Segunda tanda de ocho secuencias en un entorno no limitado

Posteriormente de Campos et al. [28] propuso un sistema de reconocimiento que usaba el modelo del color de la piel para detectar la cara, entonces usar GWN [29] para detectar puntos de referencia (ojos, nariz y boca) y rastrear esos rasgos. Para cada *frame* individual, son extraídos los *eigenfeatures* y algoritmo de selección

de rasgos es aplicado sobre la combinación de todos ellos. Finalmente un par de clasificadores y un superclasificador realizan la clasificación final del vídeo (Fig. 2.7 y Fig. 2.8). Los resultados obtenidos daban un 97.7% de precisión usando 174 imágenes de ojos de 29 personas (6 imágenes por persona).

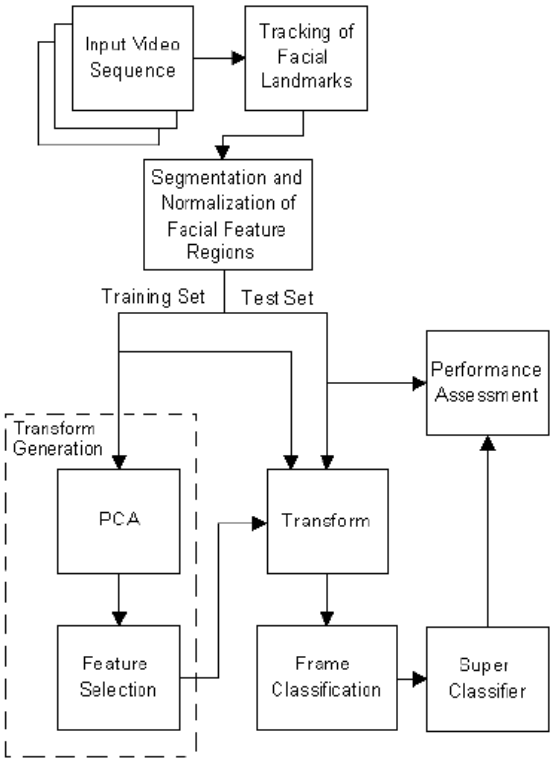

Fig. 2.7 Visión general del proyecto [28]

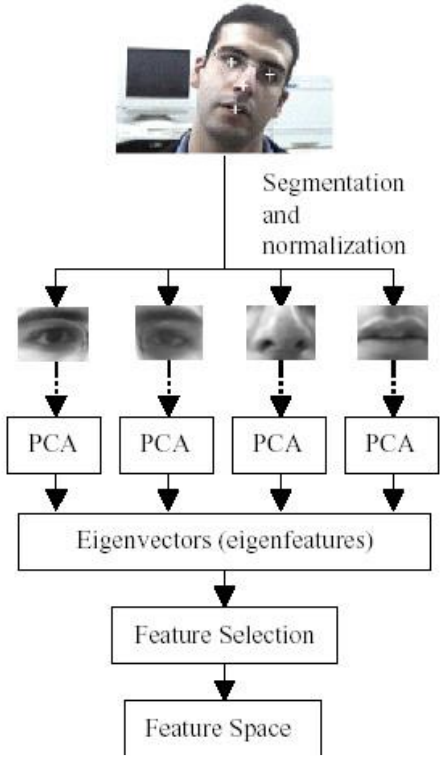

Fig. 2.8 Generación del espacio de rangos

Recientemente, algunos enfoques han utilizado un paradigma video-a-video [30] [31] en el cual la información y *frames* obtenidos de las secuencias son combinadas y asociadas con un individuo. Esta metodología implica un análisis temporal de las secuencias y una condensación de los problemas de rastreo y reconocimiento. Además, otros esquemas han incorporado información de otras modalidades para mejorar el reconocimiento facial a parte de las secuencias de video. Por ejemplo, [32] utiliza información estéreo y reporta una precisión del 90% mientras que [33] explota tanto el audio y el vídeo, como información en 3D sobre la cara para obtener una precisión del 100% para 26 sujetos.

2.1.3.2.1 Ventajas y desventajas de las secuencias de vídeo

Los sistemas de reconocimiento facial dinámicos parecen estar en desventaja frente a sus contrincantes estáticos ya que, en general, se ven obstaculizados por uno o más de los siguientes problemas: imágenes de baja calidad (aunque la calidad de imagen puede ser mejorada mediante técnicas de super-resolución, fondos desordenados, presencia de más de un rostro en la imagen y una gran cantidad de datos a procesar. Además la imagen de la cara puede ser mucho más pequeña que el tamaño requerido por la mayoría de los sistemas de reconocimiento facial por módulos.

Sin embargo, los esquemas dinámicos presentan numerosas ventajas frente las técnicas estáticas: la gran abundancia de datos permite al sistema elegir el *frame* con la mayor calidad posible y descartar los menos satisfactorios. Igualmente, estos esquemas proporcionan una continuidad temporal por lo la información puede ser combinada para mejorar el rendimiento del reconocimiento. Sin olvidar que las vídeo secuencias permiten rastrear las caras en las imágenes con grandes variaciones en las expresiones y poses [34]. También, tienen una ventaja sobre los estáticos cuando se trata de la detección de la cara en una escena ya que pueden segmentar el rostro de una persona en movimiento [35].

2.1.3.3 Otros sensores: 3D e infrarrojos

La mayoría de las investigaciones en reconocimiento facial se han centrado en identificar individuos desde imágenes de intensidad 2D. Sin embargo, en los últimos años la atención se ha dirigido a explotar otras modalidades como el 3D o imágenes de infrarrojos.

2.1.3.3.1 Modelo 3D

La razón principal a favor de utilizar información 3D para el reconocimiento facial es que permite explotar las características basadas en la forma de la cara y su curvatura (como la forma de la frente, la mandíbula y las mejillas), sin estar influidas por la iluminación, la orientación y el ruido de fondo que afecta a los sistemas 2D. Otra de los argumentos a favor de la utilización de sistemas 3D es que es la forma más directa de entrada y registro de información más compleja para el análisis. Los inconvenientes de estos enfoques son su complejidad y su coste computacional alto. Las siguientes técnicas son usadas actualmente para obtener información 3D [36]:

- Sistemas de escaneo: Escáneres faciales laser con los que se obtienen resultados muy precisos, sin embargo, el coste de estos escáneres comerciales es obviamente sustancial.

- Sistemas de luz estructurada: Estos sistemas hacen uso de los principios de la visión estero para obtener información. Su principal ventaja es que el único equipamiento necesario son algunas cámaras y algún tipo de sistema de proyección. El principal inconveniente de estos sistemas es que pueden experimentar dificultades en la resolución del patrón de la imagen tomada por la cámara.

- Sistemas de visión estero: Estos sistemas extraen información 3D de dos o más imágenes 2D tomadas desde distintos ángulos. La limitación de estos sistemas porque están limitados a objetos que generan un número suficiente de rasgos para poder llegar a una conclusión.

- Representación inversa / forma de sombreado: Estas técnicas intenta construir la forma de un objeto utilizando el conocimiento sobre la iluminación y las propiedades físicas del objeto.

Muchos enfoques han propuesto integrar la textura 2D y la información de la forma 3D, imágenes de intensidad PCA [37], perfiles de intensidad [38] o técnicas ICP [39] (iterative closest point), Gabor wavelets o análisis local de rasgos [40].

2.1.3.3.2 Infrarrojos

Desde que las imágenes de infrarrojos son relativamente insensibles a variaciones lumínicas, se han convertido en una opción para detectar y reconocer caras. De hecho, las imágenes tomadas con infrarrojos revelan las venas y la estructura del tejido facial que es única en cada individuo, como las huellas dactilares [41]. Sin embargo, existen una multitud de factores que desalientan la explotación de este tipo de imágenes para el reconocimiento facial, como por ejemplo, lo costosos que son los sensores termales, el hecho de que la radiación infrarroja puede ser desviada por el vidrio (haciendo posible ocultar parte de la cara utilizando gafas) y, por último, las imágenes de infrarrojos son sensibles a cambios de temperatura, viendo o cambios metabólicos en el sujeto.

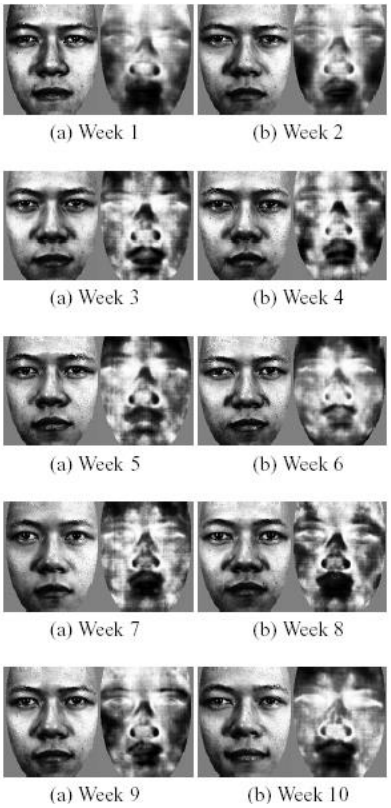

Fig. 2.9 Imágenes normalizadas tomadas a un sujeto mediante infrarrojos a lo largo de 10 semanas

El efecto del tiempo sobre los cambios faciales [42] puede apreciarse en la (Fig. 2.9) dónde el resultado del algoritmo *eigenfaces*, en ambas modalidades, era el mismo en imágenes tomadas en la misma sesión en lapsos de tiempo cercanos, mientras que en imágenes tomadas en distintas sesiones podemos apreciar cambios significativos.

2.2 Casos de uso de sistemas de seguridad con reconocimiento facial a gran escala

El impulso de la tecnología de reconocimiento facial ha provocado que surjan gran cantidad de aplicaciones en numerosos ámbitos de la seguridad: vigilancia, investigación de crímenes o control de grandes masas de gente. La no intrusividad y los bajos costes para utilizar estos sistemas en algunas de sus modalidades han motivado su aparición. No obstante, los resultados obtenidos por estos sistemas en algunos casos no han sido satisfactorios debido a la todavía corta edad en su momento del reconocimiento facial. En los siguientes apartados presentaremos algunos de los casos de uso de estos sistemas que han sido empleados con anterioridad frente a un gran número de personas.

2.2.1 Smart CCTV en Newham

CCTV (Closed-circuit televisión) es un sistema de seguridad que utiliza cámaras de vídeo para transmitir un señal a un lugar específico o a un conjunto de monitores. Con el avance de la tecnología y que la vigilancia se ha convertido en una parte de la vida, apareció la nueva generación de CCTV, las Smart CCTV.

Fig. 2.10 Oficial de policía frente los monitores de las Smart CCTV

Las Smart CCTV son la combinación de video vigilancia junto con la tecnología de reconocimiento facial para buscar, reconocer y rastrear rostros [43]. Los primeros en implantar este tipo de sistemas fueron los británicos en el año 2001 en el distrito londinense de Newham. El sistema fue enlazado a la sala de control de operaciones central de Las Fuerzas Policiales Metropolitanas de Londres (Fig. 2.10). Utilizando este sistema y la gran cantidad de cámaras de seguridad pretendían rebajar el crimen un 10% en los 6 primeros meses. La gente era rutinariamente escaneada por ambos sistemas y sus caras eran comparadas con las bases de datos de la policía. Poco tiempo después se puso en marcha este sistema en Birmingham también.

Pese a los grandes esfuerzos de desplegar un sistema de seguridad con reconocimiento facial, los resultados fueron decepcionantes [44]. Desde el año 2004, el sistema no ha conseguido reconocer ni a un solo criminal a pesar de tener una gran base de datos con todos los criminales de la zona. Sin embargo, esta información puede estar en conflicto con los datos, ya que si existió una reducción de la tasa de criminalidad. Esto se puede deber a que cuando el público sabe que está constantemente bajo video vigilancia, el propio miedo puede reducir solo la tasa de criminalidad. Esto nos lleva a la conclusión de que esa tecnología no funcionaba particularmente bien, pero la percepción del usuario de ella si lo hace [45].

2.2.2 Super Bowl XXXV en Florida 2001

La trigésima quinta edición de uno de los mayores espectáculos deportivos en EE.UU, la Super Bowl, fue el escenario de uno de los mayores experimentos con sistemas de reconocimiento facial [46]. Durante el partido, los rostros de los 100.000 aficionados que asistieron al estadio fueron analizados y comparados con las bases de datos de la policía a través de una gran cámara de vídeo. En pocos segundos cada foto era digitalizada y comparada electrónicamente frente a estas bases de datos que incluían criminales menores, carteristas e incluso terroristas en busca y captura por el FBI. El software que se utilizó para realizar el reconocimiento facial pertenecía a la

empresa Viisage y su coste para la policía fue de 3.5 millones de dólares con la promesa de que sería una gran ventaja contra los criminales [47].

Finalmente, la policía localizó a 19 rastros que coincidían con las fotos de personas arrestadas en el pasado por ser carteristas y otros cargos menores. Sin embargo, estos resultados estaban muy lejos sus expectativas. Estos resultados provocaron críticas a los departamentos de policía tanto por el alto coste del sistema como desde las agencias de protección de datos y privacidad de las personas [47].

Este sistema de seguridad quería ser implantado en otros eventos deportivos como los juegos olímpicos de invierno de Salt Lake City en 2002 o para controlar la plaga de *hooligans* en los estadios de fútbol de toda Europa [48].

2.2.3 Atentado Maratón de Boston 2013

La motivación a la hora de usar sistemas de reconocimiento facial en investigaciones criminales ha ido aumentando los últimos años. En agosto del 2011, 4 días de disturbios asolaron Inglaterra. A pesar del gran despliegue de CCTV, las fuerzas de la ley fueron incapaces de identificar a los provocadores de los disturbios con la tecnología del reconocimiento facial y, de hecho, pidieron a los ciudadanos que ayudaran compartiendo sus vídeos con las agencias de noticias.

Las autoridades de EE.UU se encontraron con un dilema similar en el atentado de la Maratón de Boston el 15 de abril del 2013. A pesar de la amplia información en forma de vídeo e imágenes de los sistemas de vigilancia, cámaras, smartphones… los federales no fueron capaces de identificar a los 2 sospechosos, los hermanos Dzhokhar y Tamerlan Tsarnaev, utilizando reconocimiento facial. Los federales

tuvieron que utilizar crowdsourcing[1] para identificar a los hermanos, sospechosos del atentado de bomba y responsables del apagón de Watertown en Massachusetts provocado por un fuego intencionado.

Mientras que la tecnología de reconocimiento facial puede ser de gran ayuda a las autoridades para confirmar la identidad de criminales sospechosos, a raíz de los pobres resultados de la Maratón de Boston los medios de comunicación tendieron a enfatizar las limitaciones de la investigación realizada allí. En [49] realiza una reexaminación de las capacidades del reconocimiento facial en este ámbito utilizando 2 softwares de reconocimiento facial:

- NEC's NeoFace v3.1 [50]
- Google-owned Pittsburgh Pattern Recognition (PittPatt) v5.2.2

Para realizar el experimento se utilizaron 6 imágenes de referencia de los hermanos tomadas de artículos de prensa (Fig. 2.11), otras 5 imágenes cedidas por el FBI que habían sido tomadas por sistemas de vigilancia en el lugar del atentado (Fig. 2.12) y como base de datos 1 millón de fotografías ofrecidas por la oficina del Sheriff de Florida. NeoFace software arrojó mejores resultados que PittPatt en todas las pruebas obteniendo un acierto entre la figura 11b y la 12b tras comparar todas las imágenes. Además también se realizó una comparación con la base de datos, siendo ésta reducida de 1 millón de fotos a poco más de 130.000 gracias a diversos filtros (edad, etnia…)

Las conclusiones obtenidas tras los experimentos dan cierto aliento a los sistemas de reconocimiento facial en estas situaciones gracias a los aciertos obtenidos. Sin embargo, este acierto es dado gracias a la calidad de la foto, una pose similar y en unas condiciones de luz bastante aceptables, lo que nos lleva a decir que los factores

[1] Crowdsourcing, del inglés crowd (multitud) y outsourcing (externalización), se podría traducir al español como colaboración abierta distribuida. Consiste en externalizar tareas que, tradicionalmente, realizaba un empleado dejándolas a cargo de un grupo numeroso de personas o comunidad.

clásicos de confusión en los sistemas de reconocimiento facial automático (iluminación, pose y expresión y ruido de fondo) siguen siendo el reto de estos sistemas comerciales.

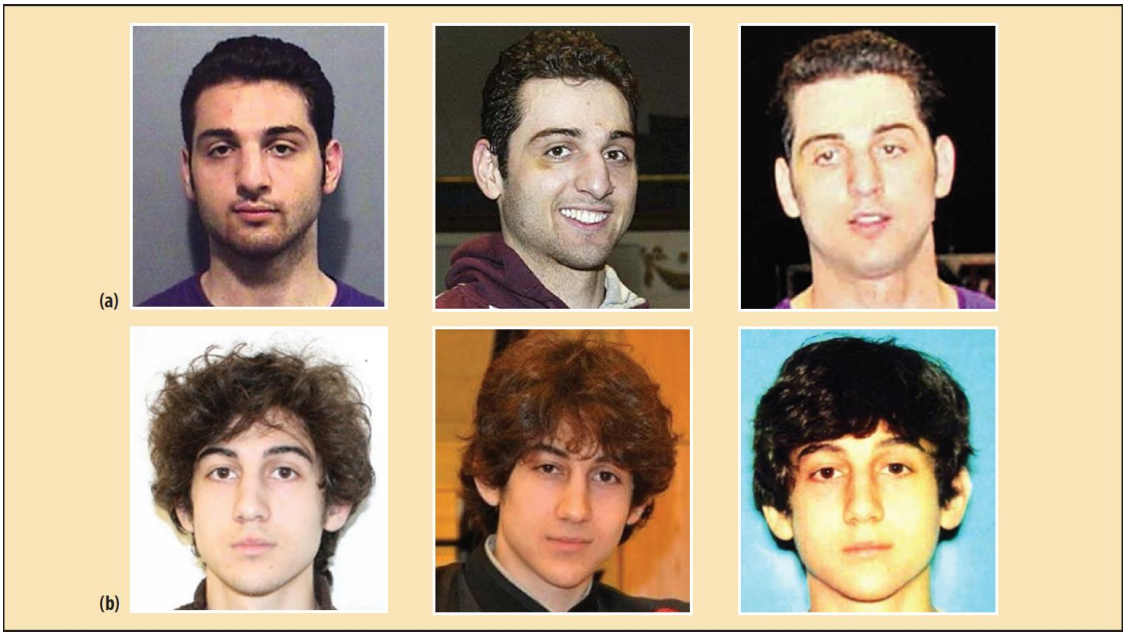

Fig. 2.11 Fotografías de los sospechosos del atentado de la Maratón de Boston cedidas por la prensa. (a) Tamerlan Tsarnaev and (b) Dzhokhar Tsarnaev

Fig. 2.12 Fotografías de los sospechosos del atentado de la Maratón de Boston cedidas por la prensa. (a) Tamerlan Tsarnaev and (b) Dzhokhar Tsarnaev

2.3 Big Data

2.3.1 Video vigilancia: El mayor de los grandes datos

Los Grandes Datos, (Big Data) continúan creciendo exponencialmente y los videos de vigilancia se han transformado en su mayor fuente. En los últimos años se han instalado una gran cantidad de cámaras de vídeo en el mundo que nos rodea, que incluyen, cámaras en los ascensores, en los cajeros bancarios, en las paredes de los edificios, en las calles, en las carreteras o incluso en nuestros ordenadores o smartphones. Todas estas cámaras capturan una cantidad de vídeos que son lanzados al ciberespacio diariamente. Londres o Pekin poseen más de un millón de cámaras instaladas por toda la ciudad. La cantidad de vídeo que capturan estas cámaras durante una hora es mucho mayor que la cantidad de programas de televisión que tiene la BBC o la CCTV (China Central Television) en sus archivos.

Un informe de la *Internation Data Corporation* [51] dice que la mitad de la Big Data, (los que son valiosos para el análisis del universo digital), fuere de videos de vigilancia en 2012 y cuyo porcentaje parece incrementarse hasta el 65% para el 2015. El gran crecimiento de estos datos han motivado numerosas actividades e investigación en el ámbito del I+D. La (Fig. 2.13) muestra el número de publicaciones en la IEEE Computer Society Digital Library y en la IEEE Xplore centrados en estos temas los diez últimos años.

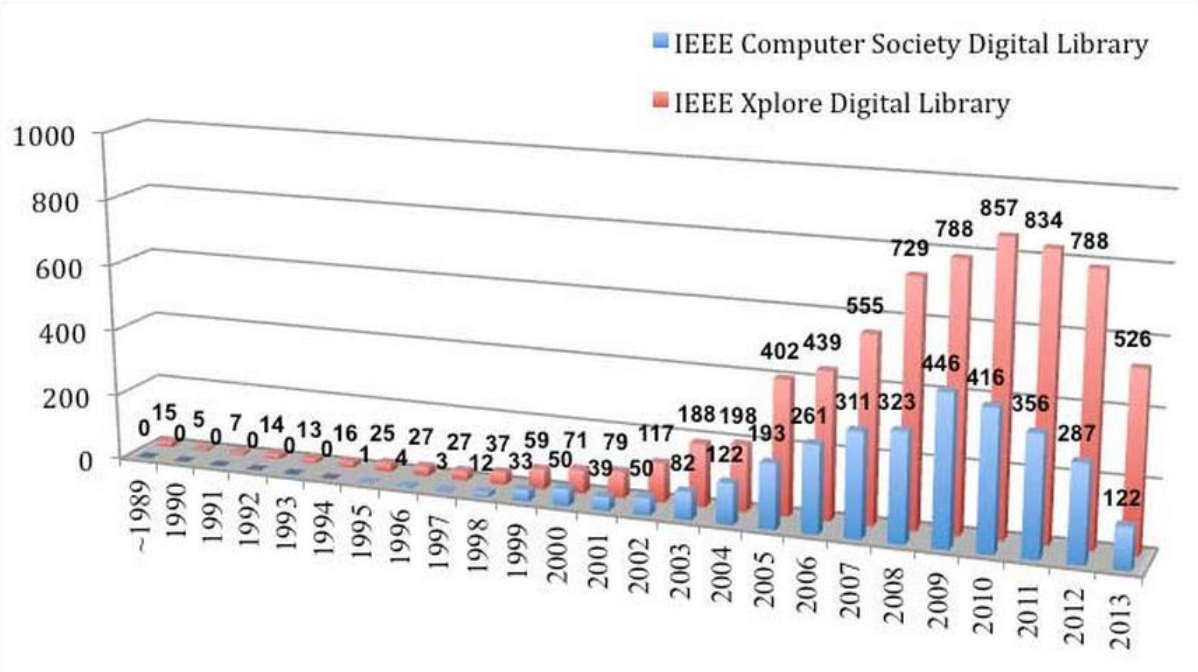

Fig. 2.13 Histograma de publicaciones en IEEE Computer Society Digital Library y en IEEE Xplore cuya metadata contiene las palabras clave video y surveillance. Los números de 2013 no están completos.

La Big Data de los videos de vigilancia presenta numerosos desafíos como la compresión de la información, el almacenamiento, la transmisión, el análisis y el reconocimiento. Respecto a este último, los seres humanos son generalmente el mayor objeto de interés en el análisis de los videos de vigilancia. La mejor ponencia presentada en la Conferencia Internacional 2013 del IEEE; “Re-Identificación de Personas basada en Referencia” [52], se propone un método basado en referencias para aprender un subespacio en el cual las correlaciones entre los datos de referencia de las diferentes cámaras se maximicen. A partir de allí, el sistema puede identificar personas que están presentes en diferentes vistas de cámaras con cambios significativos de iluminación

2.4 Cloud Computing:

2.4.1 Introducción

Cloud computing, a veces simplemente “la nube”, es la oferta de recursos de computación bajo demanda que van desde aplicaciones a centros de datos por todo internet y con el sistema de pago por uso. El cloud es la poderosa combinación de computación, comunicación en red, almacenamiento, soluciones de gestión y aplicaciones de negocio que facilitan una nueva generación de TI y servicios de consumo. Estos servicios están disponibles bajo demanda y se entregan económicamente sin compromiso de seguridad o funcionalidad.

El cloud computing se basa en la distribución de los recursos para lograr coherencia y economías de escala similares a una utilidad (como la red eléctrica) por toda la red. Es un concepto que incluye una infraestructura convergente y servicios compartidos.

Además, el cloud se centra en la maximización de la eficacia de los recursos compartidos. Los recursos cloud normalmente no solo suelen ser compartidos por varios usuarios si no que son reasignados dinámicamente según la demanda. Esto puede funcionar para la asignación de recursos a usuarios en diferentes usos horarios. Este enfoque permite maximizar el poder de computación reduciendo también el impacto medioambiental, usando menos potencia, sistemas de refrigeración y espacio en los centros de computación para las mismas funciones motivado el modelo opex (operational expenditure) que utiliza una infraestructura cloud compartida y pagas cuando la usas.

Los defensores del cloud afirman que la computación en la nube permite a las empresas evitar el coste de las infraestructuras por adelantado y tener sus aplicaciones en funcionamiento más rápido, con mayor capacidad de gestión, menos

mantenimiento y permite una mejor adaptación de los recursos para satisfacer las impredecibles fluctuaciones de la demanda.

El cloud computing es el resultado de la evolución y de las tecnologías y paradigmas existentes (Fig. 2.14). El objetivo del cloud computing es permitir a los usuarios hacer uso de todas estas tecnologías sin necesidad de tener grandes conocimientos o ser grandes expertos en ellas, permitiéndoles centrarse en los verdaderos objetivos de su negocio evitando obstáculos TI.

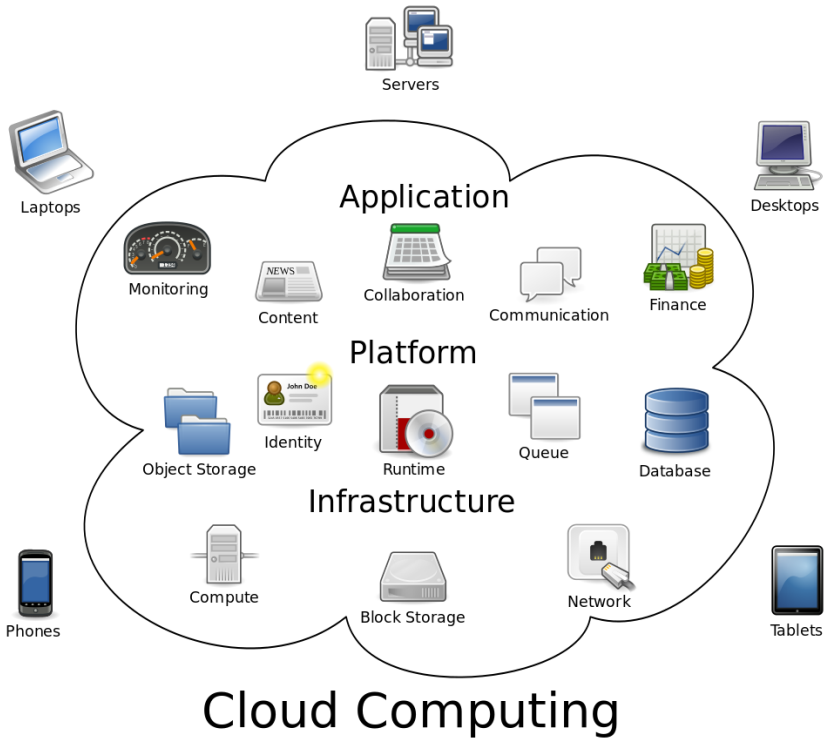

Fig. 2.14 Diagrama lógico del cloud computing

La principal herramienta tecnológica del cloud computing es la virtualización. La virtualización abstrae la infraestructura física, el cual es el componente más rígido, y la transforma en un componente software más fácil de usar y manejar. Además, la virtualización provee agilidad para mejorar las operaciones IT y reduce costes incrementando la infraestructura a utilizar.

El cloud computing además también aprovecha los conceptos de utility computing con el fin de proporcionar datos y métricas de los servicios utilizados. Además, los servicios medidos son parte esencial del ciclo de retroalimentación permitiendo servicios escalados a la carta y recuperaciones de fallos automáticas.

2.4.2 Características esenciales:

El Instituto Nacional de Estándares y Tecnología (NIST) identifica “cinco características esenciales” del cloud computing [53]:

- **Autoservicio bajo demanda**: Un consumidor puede aprovisionar unilateralmente las capacidades de computación, como el tiempo de servicio o el almacenamiento en red, de forma automática como necesite sin necesidad de interacción humana.

- **Extenso acceso a la red**: Extensas capacidades disponibles en la red y su acceso debe ser a través de mecanismos normales que proveen el uso de plataformas heterogéneas, por ejemplo, teléfonos móviles, tablets, portátiles y estaciones de trabajo.

- **Puesta en común de servicios:** Los recursos de computación de los proveedores están al día para servir a múltiples consumidores usando un modelo multi-inquilino (multi-tenant), con diferentes recursos físicos y virtuales asignados y reasignados dinámicamente de acuerdo con la demanda del consumidor.

- **Elasticidad ágil:** Los recursos pueden ser elásticamente aprovisionado y liberados, en algunos caso, de forma automática, para escalar rápidamente hacia afuera y hacia adentro con la demanda. Para el consumidor, los recursos disponibles a menudo parecen ilimitados y pueden ser apropiados en cualquier cantidad en cualquier momento.

- **Servicio monitorizado**: Los sistemas cloud controlan y optimizan automáticamente el uso de los recursos mediante el aprovechamiento de la capacidad de medición en un cierto nivel de abstracción adecuado para ese tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda, y las cuentas de usuario activas). El uso de recursos puede ser monitoreado, controlado y reportado, proporcionando transparencia tanto para el proveedor y consumidor del servicio utilizado.

2.4.3 Modelos de servicio o capas:

Los proveedores de cloud computing ofrecen servicios conforme a diferentes modelos: infraestructura como servicio (IaaS), plataforma como servicio (Paas) y software como servicio (SaaS) dónde IaaS es el más básico y cada modelo superior se abstrae de los detalles de los niveles inferiores [53].

Infrastructure as a Service (IaaS) es el servicio cloud más básico donde los proveedores ofertan computadores (físicos o más a menudo máquinas virtuales) y otros recursos. Los clouds IaaS suelen ofertar recursos adicionales como máquinas virtuales, librerías de imágenes de discos, firewalls, balanceadores de carga, direcciones IP, VLANs y paquetes de software. Los proveedores de IaaS suministran estos recursos bajo demanda a través de sus grandes piscinas instaladas en sus centros de datos. Para una conectividad amplia, los clientes pueden utilizar internet o carrier cloud (nubes portadoras) en caso de redes privadas.

Los proveedores de IaaS normalmente facturan servicios IaaS sobre una base de utility computing: el coste refleja la cantidad de recursos asignados y consumidos. Los proveedores de IaaS más típicos son: Amazon EC2, AirVM, Azure Services Platform, DynDNS, Google Compute Engine y HP Cloud.

Los proveedores de **Platform as a Service** (PaaS) normalmente ofrecen una plataforma informática que incluye sistema operativo, entorno de ejecución de lenguajes de programación, base de datos y servidor web. Los desarrolladores de aplicaciones pueden desarrollar y ejecutar sus soluciones software en una plataforma en la nube sin el coste y la complejidad de la compra y gestión de las distintas capas de hardware y software subyacente. Algunas PaaS ofrecen escalar automáticamente los recursos informáticos y de almacenamiento subyacentes para satisfacer la demanda de aplicaciones de tal forma que no exista interacción humana.

Software as a Service (SaaS) se utiliza en el modelo de negocio donde los usuarios tienen acceso a aplicaciones software y a bases de datos. Los proveedores cloud gestionan la infraestructura y las plataformas en las que se ejecutan las aplicaciones. SaaS a veces también “software bajo demanda” tiene un precio por lo general de pago por uso. Los proveedores SaaS cobran una cuota de suscripción para utilizar su software.

Los proveedores en el modelo SaaS instala y operan las aplicaciones software y los usuarios acceden como clientes. Los usuarios no gestionan la infraestructura de la nube ni la plataforma donde se ejecuta la aplicación. Esto elimina la necesidad de instalar y ejecutar la aplicación en equipos propios del usuario, lo que simplifica el mantenimiento y el apoyo técnico. Las aplicaciones en la nube son diferentes de otras aplicaciones en su escalabilidad. Los balanceadores de carga distribuyen el trabajo a lo largo del conjunto de máquinas virtuales disponibles. Este proceso es transparente para el usuario que sólo ve un único punto de acceso. Algunos ejemplos de SaaS son: Google Apps, Microsoft Office 365, TradeCard y Marketo.

El SaaS tiene la potencia para reducir el coste operacional TI a través del outsourcing del hardware y el mantenimiento del software de los proveedores de cloud. Además, las aplicaciones centralizadas pueden recibir actualizaciones sin necesidad de que los usuarios instalen nuevo software. Uno de los inconvenientes del SaaS se centra en la privacidad ya que toda la información del usuario está almacenada en el servidor del proveedor de cloud. Como resultado, podría existir un acceso no autorizado a los datos por parte de los servidores cloud.

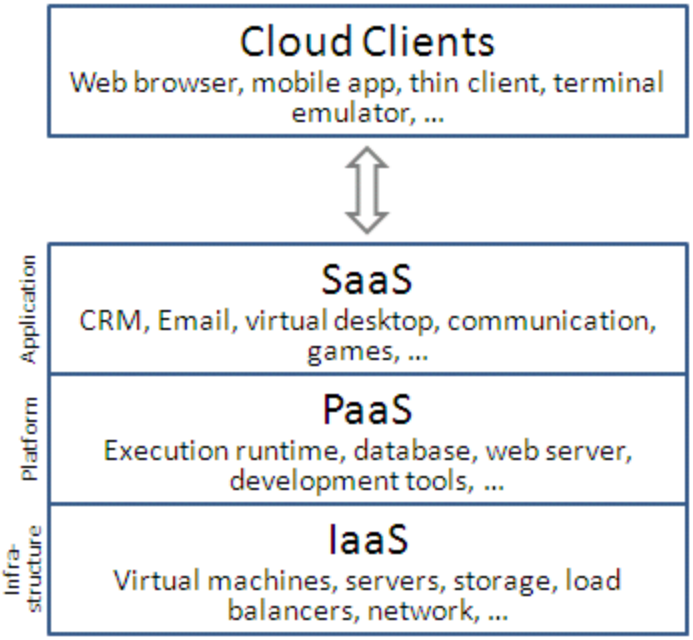

Fig. 2.15 Capas del cloud computing

2.4.4 Tipos de cloud

Existen cuatro tipos principales de modelos cloud (Fig. 2.16). En la siguiente sección se habla de los escenarios posibles de cada modelo. Estos modelos han sido recomendados por el NIST [53].

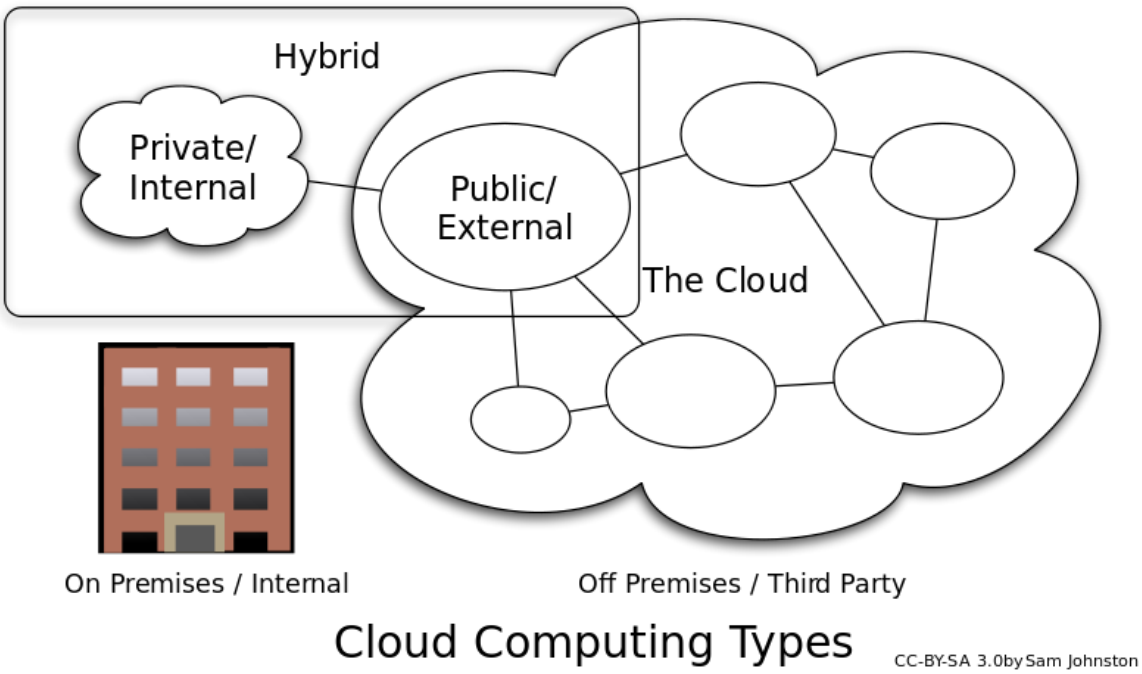

Fig. 2.16 Modelos de despliegue del cloud computing

El **cloud privado** es una infraestructura cloud que opera solamente para una única organización, y son administrados internamente o por un tercero. Realizar un proyecto en el cloud privado requiere un alto nivel de compromiso a la hora de virtualizar el entorno de negocios importantes y requiere organización para evaluar las decisiones sobre los recursos existentes. Este modelo puede mejorar el negocio pero cada paso en el proyecto plantea problemas de seguridad que se deben abordar para prevenir posibles vulnerabilidades graves.

El **cloud público**, los servicios se prestan a través de una red que está abierta para uso público. Técnicamente no hay diferencia entre la arquitectura cloud pública y privada, sin embargo, las consideraciones de seguridad puede ser sustancialmente diferentes para los servicios (aplicaciones, almacenamiento y otros recursos) que son puestos a disposición por un proveedor de servicios para una audiencia pública y cuando la comunicación es efectuada a través de una red sin confianza.

	Cloud Público	Cloud Privado
Coste inicial	Normalmente cero	Normalmente elevado
Coste de ejecución	Predecible	Impredecible
Personalización	Limitada	Posible
Privacidad	No	Sí
Log-in	Imposible	Posible
Escalado	Sencillo con límites definidos	Laborioso pero sin límites

Tabla 2.1 Modelos público y privado para SaaS

El **cloud comunitario** comparte infraestructuras entre numerosas organizaciones con preocupaciones comunes (seguridad, cumplimiento, jurisdicción, etc), y son administrados internamente o por un tercero. Los gastos se reparten entre menos usuarios que en el cloud público (pero más que en el cloud público), por lo que sólo algunos de los posibles ahorros de coste son posibles.

El **cloud híbrido** es una composición de 2 o más clouds (privado, comunitario o público) que siguen siendo entidades únicas pero están unidas ofreciendo los múltiples beneficios de los modelos desplegados. Dicha composición permite opciones de desarrollo para los servicios en la nube, permitiendo a las organizaciones TI utilizar los recursos de computación de la nube pública para satisfacer necesidades temporales. Esta capacidad permite a las nubes híbridas emplear el cloud bursting para escalar a través de los clouds. Los clouds bursting son un modelo de aplicación que se ejecuta sobre un cloud privado y se extiende hacia un cloud público cuando existen picos de demanda permitiendo a las organizaciones pagar solamente por más recursos cuando lo necesitan.

La figura (Fig. 2.17) muestra que tipo de aplicaciones puede ejecutarse bajo cada modelo de cloud:

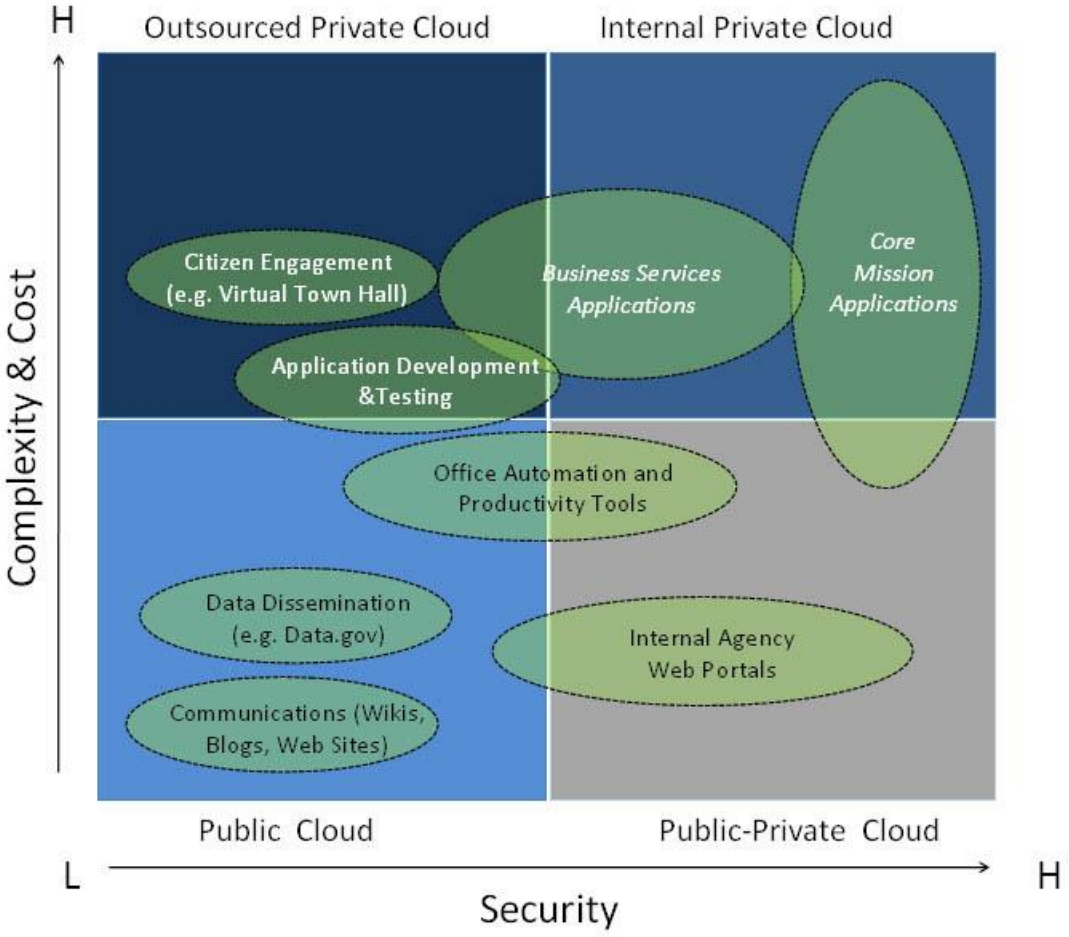

Fig. 2.17 Diferentes tipos de aplicaciones en sus respectivos tipos de cloud

2.4.5 Beneficios

El cloud computing, independientemente de su desarrollo y modelo de servicio, presenta los siguientes beneficios:

- **Agilidad:** Capacidad de mejora para ofrecer recursos tecnoógicos al usuario por parte del proveedor.

- **Coste:** Los proveedores de cloud afirman que los costes se reducen. Un modelo de cloud público convierte los gastos de capital en gastos de funcionamiento. Ello reduce barreras de entrada, ya que la infraestructura se

proporciona típicamente por una tercera parte y no tiene que ser adquirida una única vez para tareas intensivas e infrecuentes.

- **Escalabilidad y elasticidad:** Aprovisionamiento de recursos sobre una base de autoservicio en casi en tiempo real, sin que los usuarios necesiten cargas de alta duración.

- **Independencia de dispositivo o localización:** Permite a los usuarios acceder a los sistemas usando un navegador web independientemente de su localización o del dispositivo que usen.

- **Multi-tenancy (multi-inquilino):** Permite compartir recursos y costes a lo largo de una gran piscina de usuarios a través de la centralización de la infraestructura (menores costes), aumenta la capacidad de carga máxima y el uso y mejora de la eficiencia de los sistemas que a menudo no se aprovechan.

- **Rendimiento:** Los sistemas cloud controlan y optimizan el uso de los recursos de manera automática. Dicha característica permite un seguimiento, control y notificación del mismo. Esta capacidad aporta transparencia tanto para el consumidor como para el proveedor del servicio.

- **Seguridad:** Puede mejorar debido a la centralización de los datos. La seguridad es a menudo tan buena o mejor que en los sistemas tradicionales, en parte porque los proveedores son capaces de dedicar recursos a la solución de problemas de seguridad que muchos clientes no pueden permitirse el lujo de abordar.

- **Mantenimiento:** Las aplicaciones cloud tienen un mantenimiento más sencillo ya que los usuarios no necesitan realizar nuevas instalaciones en sus sistemas y pueden acceder desde diferentes lugares.

- **Virtualización:** Permite a los servidores y a los dispositivos de almacenamiento ser compartidos y su uso se ve incrementado. Las aplicaciones pueden ser fácilmente migradas de un servidor físico a otro.

2.5 Conclusión

Los beneficios y características del cloud computing como su rápido despliegue, compartición de recursos, fiabilidad, elasticidad y la reducción de costes que ofrece, lo hace muy atractivo para la migración de los servicios de las compañías. Las predicciones auguran un crecimiento del despliegue de la tecnología cloud en los próximos años de más de un 56% [54]. Sin embargo, la migración de estos servicios presenta numerosos retos como la gestión y la seguridad.

Los sistemas de seguridad basados en reconocimiento facial presentan numerosas ventajas frente a otros sistemas biométricos que necesitan amplios despliegues económicos o ponen en riesgo la salud de los individuos. Debido a su buen funcionamiento estos sistemas han sido muy utilizados como herramientas de identificación en entornos con las variables que podrían condicionar el rendimiento de estos sistemas (luz, pose, expresión o ruido de fondo). Sin embargo, la aplicación de estos sistemas de seguridad a otros tipos de escenarios más dinámicos como las Smart CCTV de Newham en Londres o la prueba en la Super Bowl del 2001 utilizando video vigilancia han presentado resultados menos satisfactorios, debidos principalmente a la imposibilidad de controlar estar variables.

La gran cantidad de datos generada por la video vigilancia ha provocado que este problema de seguridad se convierta en un problema de Big Data, siendo más de la mitad de la Big Data global para el año 2015. Para manejar esta incontrolable cantidad de información se están estudiando numerosas soluciones para los problemas de almacenamiento, compresión, gestión y potencia de ejecución

necesaria para utilizar esos datos. Una gran cantidad de información y de datos permitiría a los sistemas de reconocimiento facial experimentar un incremento en su rendimiento siempre y cuando se sea capaz de gestionar tal cantidad de información.

En el siguiente capítulo es dado el modelo de un sistema de seguridad basado en el reconocimiento facial desplegado sobre un cloud público y con potencia suficiente para hacer frente a grandes flujos de datos, permitiendo la comparación con grandes bases de datos. Con este modelo se pretende mejorar el rendimiento de estos sistemas de seguridad, abaratar sus costes gracias al pago por uso del cloud público y hacer frente a grandes picos de computación en intervalos de tiempo cortos.

Capítulo 3

3. Propuesta

El principal problema de los sistemas de reconocimiento facial a gran escala es el flujo de datos que entra al sistema. Se necesita una gran potencia de computación para hacer frente a “tiempo o real” o “semi-real” a este flujo de dados. Otro problema de estos sistemas es el control de las variables del entorno para maximizar su rendimiento.

Los modelos de arquitectura para el sistema de seguridad basado en reconocimiento facial son: **Sistema de Reconocimiento Facial Cloud para escenarios estáticos** y **Sistema de Reconocimiento Facial Cloud para escenarios dinámicos.** El primero consiste en recrear un gran flujo de datos de imágenes de individuos tomadas de frente bajo un entorno controlado y compararlas con las grandes bases de datos de los organismos de seguridad mundiales (FBI e Interpol), dentro de un límite de tiempo, el cual quedará establecido por el tiempo que tarda un individuo en abandonar la zona controlada por las autoridades. El segundo consiste en tomar imágenes cada cierto intervalo de tiempo a través de cámaras IP situadas en lugares estratégicos tomando imágenes mucho más dinámicas de los usuarios y con las variables del entorno no controladas.

El sistema de reconocimiento facial será implementado utilizando **Luxand FaceSDK** que utiliza el método de reconocimiento de *featured-based* de alto rendimiento. Utilizando este SDK implementaremos tres programas, uno para cada fase de del reconocimiento facial: reconocimiento de la cara (*recognition*), extracción de los rasgos faciales (*facefeatures*) y comparación (*matching*).

Para analizar la viabilidad del sistema, se medirá tanto el tiempo de comparación de una foto con una base de datos de un tamaño específico, en el caso del matching, como el tiempo de reconocimiento de la cara y extracción de rasgos en los otros casos. Posteriormente se extraerán las ecuaciones de cada máquina para poder dar un análisis más completo de su rendimiento y compararlo con el coste necesario para asumir ese despliegue.

Las instancias que se utilizarán para realizar los experimentos se muestran en la siguiente tabla:

Nombre Instancia	vCPU	ECU	Memoria RAM (GiB)	Almacenamiento (GB)	Plataforma	Rendimiento de la red
t1.micro	1	1 o 2 (ráfagas)	0.613	Ninguno (Sólo Amazon EBS)	32/64 bits	Muy bajo
m1.small	1	1	1.7	1x160	32/64 bits	Bajo
m1.large	2	4	7.5	2x420GB	64 bits	Moderado
m1.xlarge	4	8	15	4x420	64 bits	Alto
m3.large	2	6.5	7.5	1x32 SSD	64 bits	Alto

Tabla 3.1 Instancias de Amazon Web Services EC2 utilizadas por los modelos del sistema de reconocimiento facial propuestos

Amazon Web Services EC2 (AWS EC2) será usado como plataforma de cloud computing. AWS Ec2 es una plataforma flexible con múltiples opciones de configuración como el sistema operativo, el tipo de instancias, la memoria de estas y el rendimiento de la red. En nuestro caso el sistema operativo en el que ejecutaremos nuestro sistema de seguridad será Linux distribución Ubuntu 12.04.

PARTE II

Experimentos y Resultados

Capítulo 4

4. Arquitectura del Sistema

Tanto la arquitectura del Sistema de Reconocimiento Facial Cloud para escenarios estáticos como la de escenarios dinámicos se establecen en el cloud público de *AWS EC2* y lanzadas desde la zona *US-cost-1d*. Todas las instancias corren el sistema operativo Ubuntu[2] 12.04 de 64 bits. Se deben tomar algunas consideraciones sobre el desarrollo con Luxand FaceSDK en su versión 5.0 [55]:

- Luxand FaceSDK es una librería de detección y reconocimiento facial que ofrece una Api para detectar y rastrear rostros, reconocer el género tanto en imágenes como en vídeo con soporte para cámaras IP.

- El SDK proporciona las coordenadas de 66 puntos faciales (Fig. 4.1) (incluyendo ojos, cejas, boca, nariz y contorno de la cara) los cuales pueden ser utilizados para el reconocimiento facial.

- Está disponible para plataformas de 32 y 64 bits como para los sistemas operativos Windows, Linux, MacOS X, iOS y Android y se puede desarrollar en los siguientes lenguajes de programación: .NET (C# y VB), Dephi, Java, Cocoa, VisualBasic 6.0, iOS y Android.

- Los requisitos mínimos son: procesador de 1 GHz y 256 MB RAM. Sin embargo los requisitos recomendados son: Procesador Intel Corel i7 o Xeon y 2GB RAM.

[2] Ubuntu es un sistema operativo basado en la distribución de Linux Debian y distribuido gratuitamente como un software open source, usando su propio entorno de escritorio. Desde 2012, de acuerdo con los estudios online, Ubuntu es el sistema operativo basado en Linux más popular para ordenadores de mesa y portátiles, y la mayor cobertura de Ubuntu se centra en su uso en ese mercado. Sin embargo, también es muy popular en los servidores y en el cloud computing.

- Soporta una rotación de la cabeza de -30..30 grados dentro del plano de rotación y de -30..30 fuera del plano de rotación.

- Toda la información detectada se devuelve en forma de coordenadas (x,y). El array de las 66 coordenadas forman un template cuyo tamaño es de 13kb.

- Posibilidad de comparar dos caras dando un FAR (*False Acceptance Rate*) y un FRR (*False Rejection Rate*).

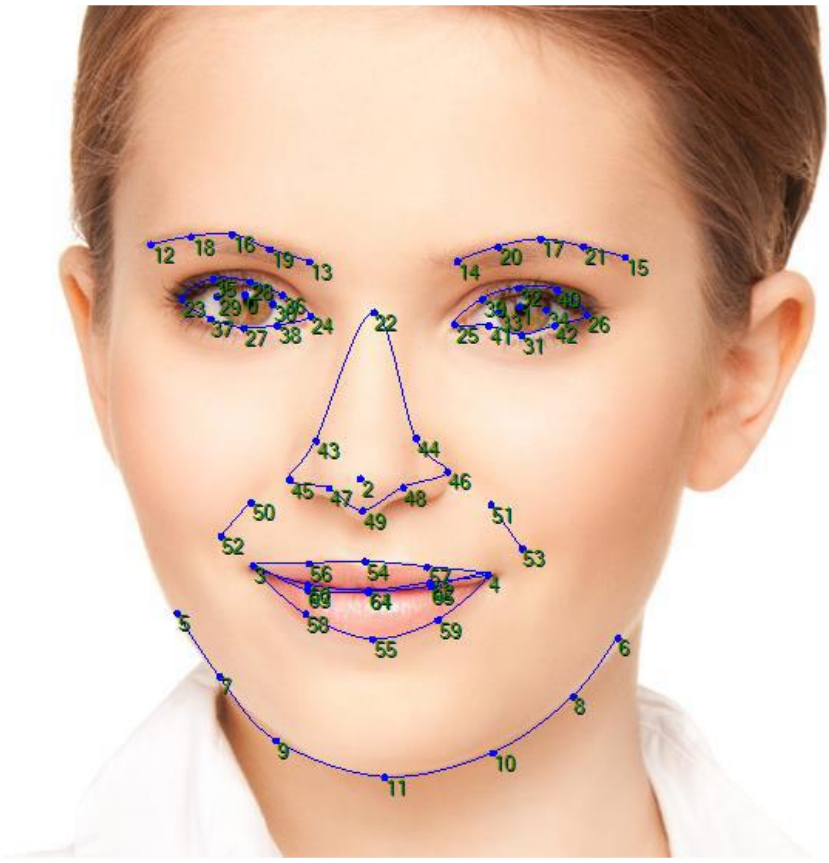

Fig. 4.1 Luxand FaceSDK detecta 66 puntos faciales

4.1 FaceSDK en las Instancias de AWS EC2

Las instancias utilizadas en los modelos anteriormente mencionados presentan una configuración específica para poder ejecutar el software desarrollado con la API de Face SDK de una forma correcta. En nuestro caso, el software se ha desarrollado utilizando el lenguaje de programación **Java**, por lo que es necesario instalarlo en nuestras instancias.

Todas las instancias utilizar el sistema operativo Ubuntu 12.04 por lo que la configuración se realizará por línea de comandos utilizando el siguiente esquema:

- Obtener permisos de superusuario con "*sudo su*" y posteriormente introducir la contraseña.
- Actualización de la instancia con "*apt-get update*" y "*apt-get upgrade*"
- Instalación del openjdk-6-jre de java con "*apt-get install openjdk-6-jre*"

Una vez hechas estas configuraciones el software para el reconocimiento facial puede ser instalado sin problemas.

El software utilizado en estos modelos se divide en tres programas: SavePicInAFile (detección), FaceFeatures (extracción de rasgos) y Match (comparación y reconocimiento).

4.2 Arquitectura del Sistema de Reconocimiento Facial Cloud para Escenarios Estáticos

Este primer modelo afronta los problemas de computación, rendimiento y fiabilidad de los sistemas de seguridad que utilizan reconocimiento facial utilizando la tecnología del cloud público. El software de reconocimiento facial será lanzado desde el cloud público de AWS EC2 como se ha mencionado antes.

Es imprescindible que las instancias utilizadas estén configuradas correctamente y que el software de reconocimiento facial esté instalado en ellas. Las fotografías de los individuos serán tomadas desde ciertos puntos de seguridad donde se encontraran cámaras conectadas a un ordenador con conexión a internet. Estas fotografías serán tomadas de frente logrando un control sobre las posibles variables del entorno. Las fotografías serán procesadas por el programa SavePicInAFile del cual obtendremos una fotografía normalizada del rostro de los usuarios. Posteriormente, se establecerá una conexión segura SFTP para enviar a las instancias del cloud la fotografía encriptada.

Para la administración del cloud vamos a utilizar una herramienta llamada StarCluster [56] que nos facilita la construcción, configuración y la gestión de clusters virtuales de máquinas EC2 de Amazon. Esta máquina no administra automáticamente el aumento o el decremento del número de instancias según el flujo datos, solamente sirve como herramienta de gestión. Una instancia llamada “master” será la encargada de distribuir el trabajo entre todas las instancias habilitadas.

Una vez en las instancias del cloud, se procederá a la extracción de los rasgos faciales de la fotografía del individuo y su posterior comparación con el resto de fotografías de las bases de datos de los diferentes organismos de seguridad. Estas fotografías se almacenarán en volúmenes EBS compartidas con todas las instancias por NFS desde la máquina master. Tras el reconocimiento facial, si el resultado es

positivo frente a alguno o algunos de los individuos almacenados en la base de datos, se transmitirá una señal de alerta al centro de seguridad donde se tomarán las medidas oportunas.

El esquema de la arquitectura del Sistema de Reconocimiento Facial Cloud para escenarios estáticos está ilustrado en la Fig. 4.2.

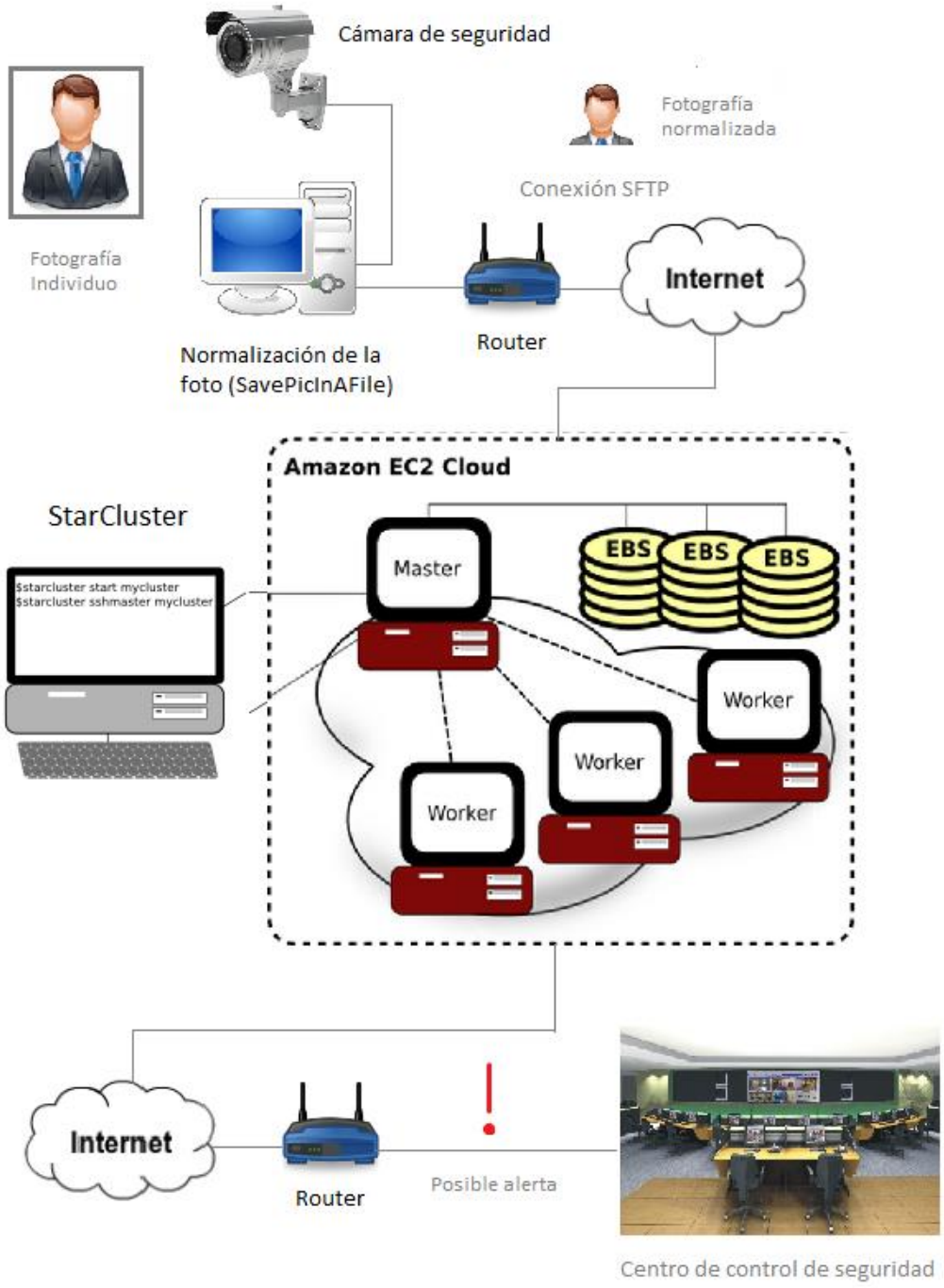

Fig. 4.2 Arquitectura del Sistema de Reconocimiento Facial Cloud para escenarios Estáticos

4.3 Arquitectura del Sistema del Reconocimiento Facial Cloud para Escenarios Dinámicos

Este modelo afronta igual que el anterior los problemas computación, rendimiento y fiabilidad en los sistemas de seguridad que utilizan reconocimiento facial pero en un entorno más dinámico como es la video vigilancia. Igual que en el modelo anterior, el software de reconocimiento facial será lanzado desde instancias del cloud público de AWS EC2.

Es imprescindible que las instancias utilizadas estén configuradas correctamente y que el software de reconocimiento facial esté instalado en ellas. Las fotografías serán tomadas por cámaras de video vigilancia colocadas en lugares estratégicos con gran número de personas para captar al mayor número de individuos al mismo tiempo y con la mayor calidad posible. De esta forma, no es posible controlar las variables de entorno de la misma forma que en el modelo anterior pero se aumenta la cantidad rostros que se pueden captar en menos tiempo. Se tomará una fotografía en cada una de las cámaras conectadas al sistema cada cierto intervalo t, que será mayor o menor dependiendo del flujo de personas y el movimiento de estas por el lugar. Inmediatamente el sistema extraerá el mayor número de caras de la fotografía con el programa SavePicInAFile, se normalizarán y se enviarán a través de internet con una conexión SFTP encriptada para mantener la seguridad y privacidad de éstas.

Para la administración del cloud vamos a utilizar una herramienta llamada StarCluster [56] que nos facilita la construcción, configuración y la gestión de clusters virtuales de máquinas EC2 de Amazon. Esta máquina no administra automáticamente el aumento o el decremento del número de instancias según el flujo datos, solamente sirve como herramienta de gestión. Una instancia llamada “master” será la encargada de distribuir el trabajo entre todas las instancias habilitadas.

Una vez en las instancias del cloud, se procederá a la extracción de los rasgos faciales de la fotografía del individuo y su posterior comparación con el resto de fotografías de las bases de datos de los diferentes organismos de seguridad. Estas fotografías se almacenarán en volúmenes EBS compartidas con todas las instancias por NFS desde la máquina master. Tras el reconocimiento facial, si el resultado es positivo frente a alguno o algunos de los individuos almacenados en la base de datos, se transmitirá una señal de alerta al centro de seguridad donde se tomarán las medidas oportunas.

El esquema de la arquitectura del Sistema de Reconocimiento Facial Cloud para escenarios estáticos está ilustrado en la Fig. 4.3.

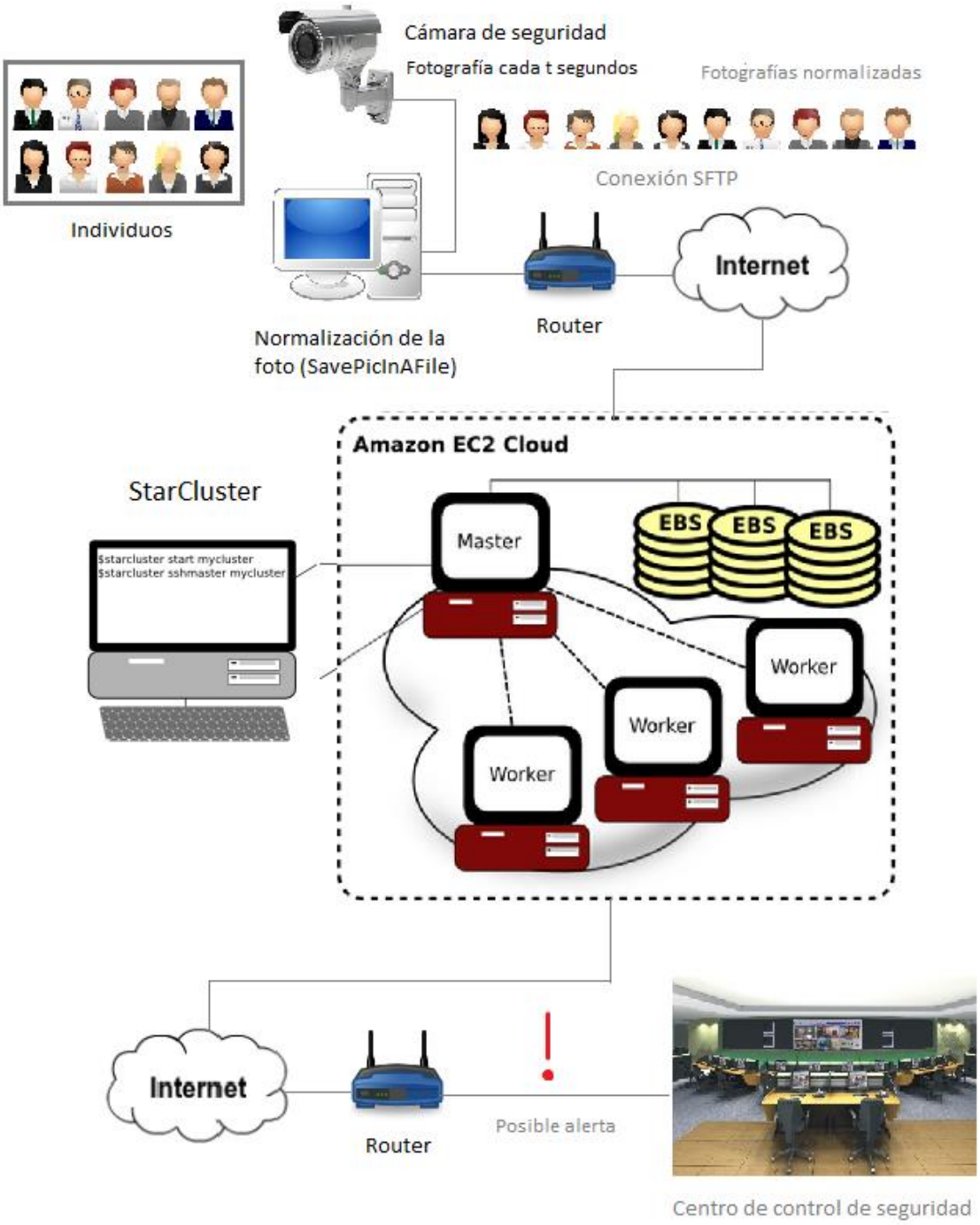

Fig. 4.3 Arquitectura del Sistema de Reconocimiento Facial Cloud para escenarios Dinámicos

Capítulo 5

5. Experimentos

Para testear las arquitecturas propuestas y poder obtener conclusiones sobre ellas en términos de rendimiento, coste y potencia de computación, se debe definir una metodología.

5.1 Metodología

Se ha utilizado la misma metodología de experimentación para ambas arquitecturas ya que en términos de computación son iguales. Para evaluar el rendimiento y los costes de las arquitecturas se han realizado pruebas de rendimiento sobre los tres programas utilizados en el sistema de reconocimiento facial:

- **SavePicInAFile**: Detecta todos los rostros de una fotografía y los guarda en imágenes independientes de un tamaño y resolución determinados.

- **FaceFeatures**: Dado la imagen de un rostro extrae de ella las coordenadas de los 66 puntos faciales guardándolos en una estructura llamada template.

- **Match**: Dados dos templates de dos fotografías realiza una comparación dando un tanto por ciento de similitud.

Estas pruebas se han realizado sobre instancias lanzadas desde el cloud público de AWS EC2 mostradas en la tabla 3.1.

Se han utilizado bases de datos de fotografías de diferentes tamaños; 100, 200, 250 y 500 fotos para las instancias con menos potencia como la t.1 micro o la m1.small mientras que para las instancias con más potencia además se han realizado pruebas con bases de datos de 2000, 5000 y 10000 fotografías. En la experimentación no se utilizaron volúmenes EBS para almacenar las fotografías si no que se cargaron directamente en las instancias. Para subir las fotografías se utilizó el siguiente comando:

Scp –r –i cdhit-key.pem rutaCarpetaFotos ubuntu@ec2-54-81-82-70.compute-1.amazonaws.com:/home

Durante las pruebas de los tres programas frente a las distintas bases de datos se recogieron diferentes datos. En el caso del programa SavePicInAFile, el tiempo medio de detección de un rostro en una fotografía y su posterior almacenamiento y normalización de ese rostro en otra imagen. Para ejecutar este programa se necesitan las fotografías tomadas a los usuarios sin normalizar.

En el caso del programa FaceFeatures, se midió el tiempo medio de extracción de los 66 rasgos faciales y su posterior almacenamiento. Para ejecutar este programa son necesarias las fotografías normalizadas de los usuarios, resultantes de ejecutar el programa SavePicInAFile.

Para el programa de comparación Match se midió el tiempo medio de comparación de una fotografía frente a todas las fotografías almacenadas en una base de datos. Para obtener una media fiable se optó por una comparación de todas las fotografías de la base de datos utilizada con todas ellas, es decir, si la base de datos era de 500 fotografías se realizaron 500x500 comparaciones. Para ejecutar este programa es necesario la fotografía normalizada del usuario a comparar, y la base de datos con las fotografías normalizadas con sus respectivos templates.

En los tres casos, también se midió el tiempo total de ejecución, sin embargo, en el caso del programa Match frente a bases de datos grandes (2000,5000 y 10000) solamente se realizaron 50 comparaciones (2000x50, 5000x50 y 10000x50) ya que los tiempos de ejecución comenzaban a ser muy altos.

En el caso de las instancias con más de una CPU y el programa Match se realizaron pruebas lanzado a ejecución el mismo programa en varias CPU a la vez para intentar obtener cual es la distribución de la carga de computación más óptima en cada instancia. Estos experimentos se realizaron utilizando un script que ejecutaba al mismo tiempo varios programas. Por ejemplo, en el caso de la máquina m1.xlarge con 4 CPU el script para ejecutar un programa Match en cada procesador tenía la siguiente forma:

scriptMatchQuadParcial.sh

```
1  #!/bin/bash
2  # -*- ENCODING: UTF-8 -*-
3  java -jar Proyecto/Match.jar $1 P > Proyecto/Resultados/resultado1 &
4  java -jar CProyecto/Match.jar $2 P > Proyecto/Resultados/resultado2 &
5  java -jar AProyecto/Match.jar $3 P > Proyecto/Resultados/resultado3 &
6  java -jar BProyecto/Match.jar $4 P > Proyecto/Resultados/resultado4 &
7  exit
```

Fig. 5.1 Ejemplo script para el programa Match ejecutando 4 tareas a la vez

Dónde $1, $2, $3 y $4 eran los parámetros que se le pasaban al script que indicaban la ruta de almacenamiento de los templates de las fotografías almacenadas en la base de datos. Era necesario realizar esta separación para evitar problemas de accesos múltiples los cuales no existirían en la arquitectura real gracias a StarCluster. El parámetro P quiere decir que se va a realizar una ejecución parcial de solo 50 comparaciones.

En estas máquinas se probaron todas las combinaciones, es decir, en el ejemplo de la máquina m1.xlarge se realizaron las siguientes pruebas:

- 1 tarea para 4 CPU
- 2 tareas para 4 CPU
- 3 tareas para 4 CPU
- 4 tareas para 4 CPU

De esta forma se puede encontrar la mejor distribución de computación para cada máquina.

Por último, los datos de la ejecución se guardan en unos ficheros txt para su posterior estudio. Además la ejecución se ha podido realizar sin necesidad de estar constantemente conectado a las instancias gracias a una *screen*[3] que mantenía la conexión abierta sin necesidad de mantener el ordenador encendido.

Cada prueba ha sido repetida mínimo dos veces para proporcionar la mayor fiabilidad a los resultados.

[3] Las instancias de AWS EC2 detienen su ejecución si la máquina desde la que se ha conectado rompe la conexión. Gracias a una screen es posible mantener esta conexión abierta permitiéndonos un ahorro energético y de tiempo permitiéndonos aprovechar las horas de la noche para ejecutar los programas. Para crear una screen se utiliza el comando *screen* en la consola, para resumirla *Ctrl*+*D* y de nuevo para conectar con ella el comando *screen –r*.

Capítulo 6

6. Resultados

Las arquitecturas propuestas se han testeado utilizando la metodología descrita en las secciones previas. Cada uno de los tres programas ha sido probado en las distintas instancias y con los resultados se ha podido extraer la ecuación del rendimiento de las máquinas.

6.1 SavePicInAFile

El programa SavePicInAFile encargado de la detección del rostro y su posterior normalización en una imagen ha sido probado en las diferentes instancias de AWS EC2 mencionadas anteriormente. Se han utilizado los resultados para construir unos gráficos de los cuales hemos podido extraer la ecuación del rendimiento de cada máquina.

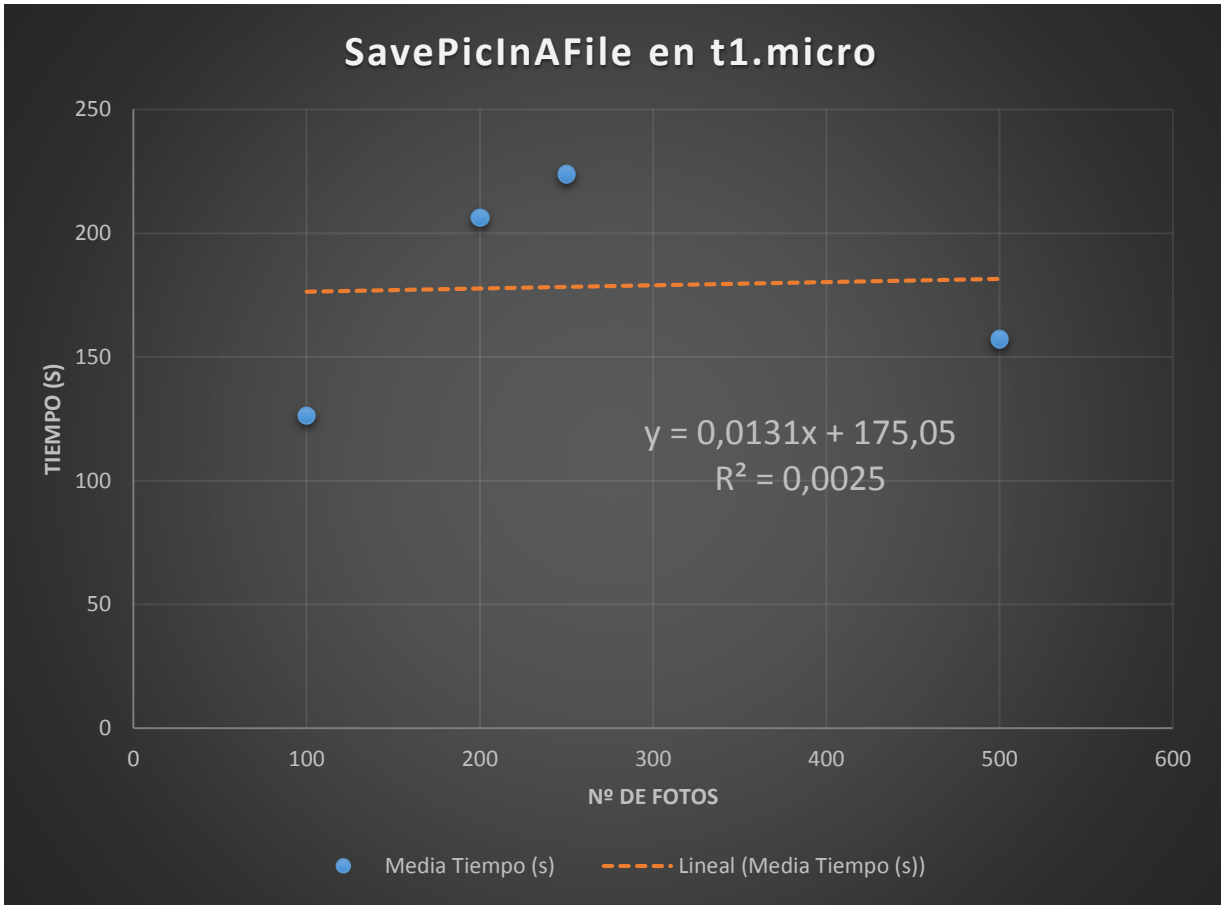

Fig. 6.1 Gráfica SavePicInAFile en t1.micro

En el gráfico (Fig. 6.1) podemos apreciar el comportamiento totalmente aleatorio de las instancias t1.micro. Esto se debe a que estas instancias funcionan “a ráfagas” utilizando la potencia de computación sobrante de otras máquinas más potentes. Esto se puede comprobar observando el tiempo necesario para detectar y normalizar 500 fotografías es inferior en este caso que el tiempo necesario para 250 fotografías.

Los problemas de rendimiento que presentan las instancias t1.micro las convierten en poco fiables para tareas que tienen que ser ejecutadas en un tiempo específico pero pueden ser útiles en otros ámbitos. La ecuación extraída de los resultados no es útil por el motivo anterior.

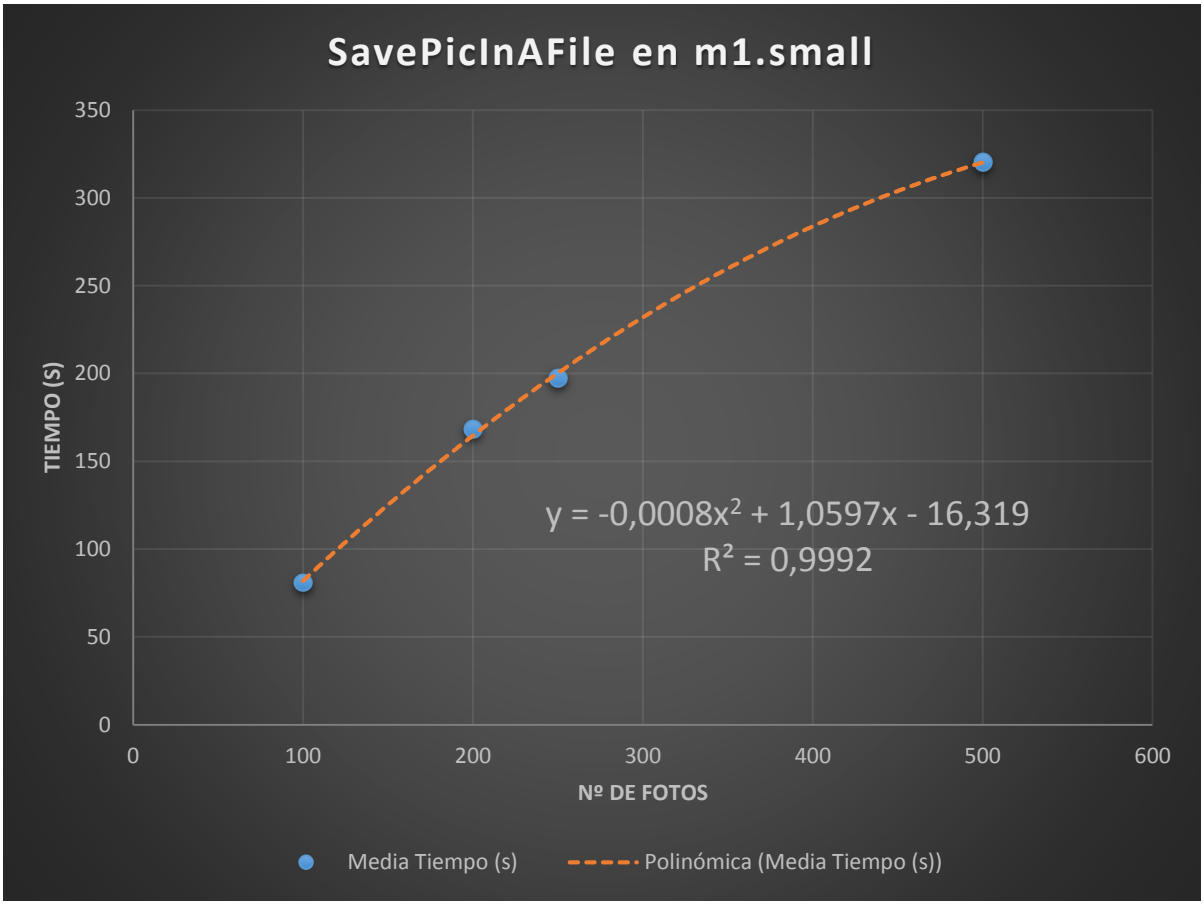

Fig. 6.2 Gráfica SavePicInAFile en m1.small

En el gráfico (Fig. 6.2) podemos apreciar el mejor comportamiento de la máquina m1.small frente a la t1.micro. En este caso los resultados no son aleatorios y mantienen una coherencia. La ecuación extraída nos permite conocer el rendimiento de la máquina entre 1 y 500 fotos.

La máquina m1.small es perfecta para tareas que no requieren gran potencia de ejecución. En el caso del programa SavePicInAFile puede detectar y normalizar 500 fotografías en 320 segundos.

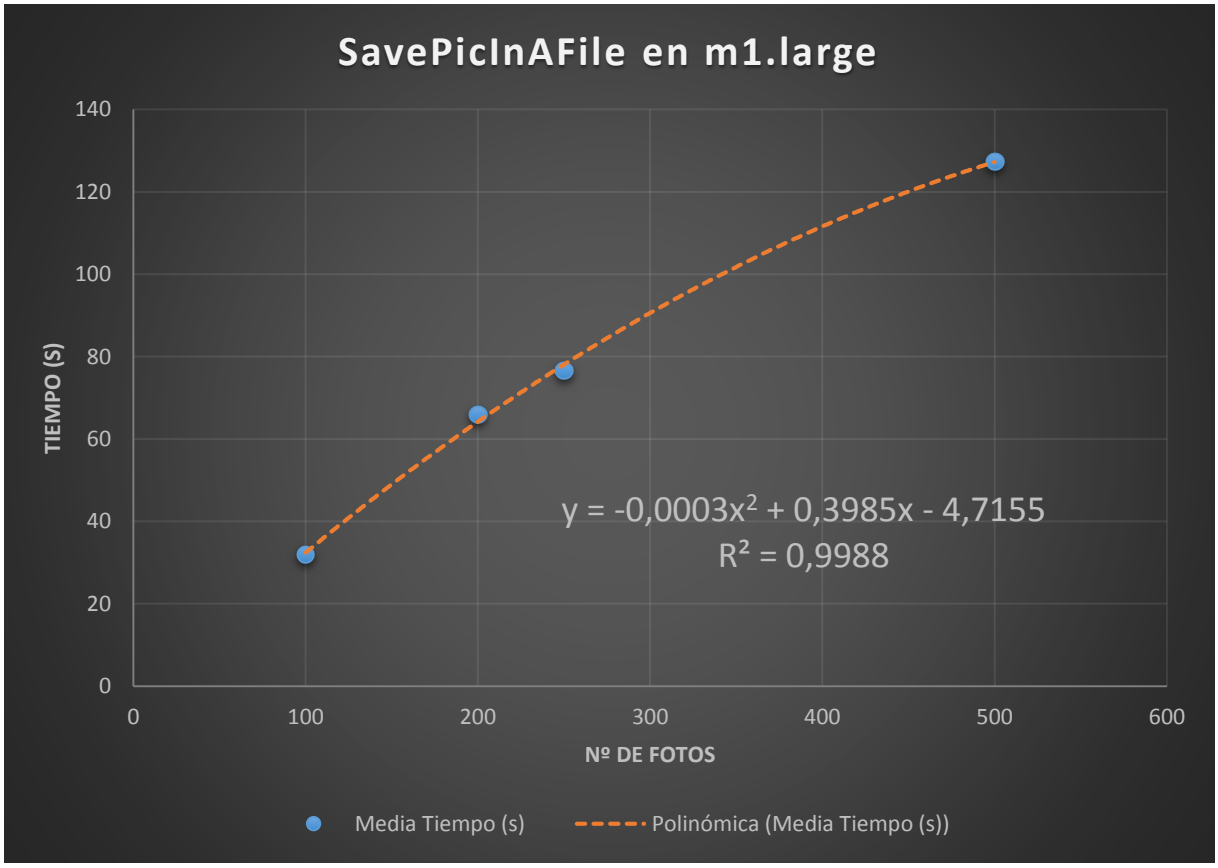

Fig. 6.3 Gráfica SavePicInAFile en m1.large

En el gráfico (Fig. 6.3) podemos ver el rendimiento del programa SavePicInAFile en la máquina m1.large. El rendimiento supera al de la m1.small gracias a los dos procesadores que posee esta máquina frente al único procesador de la máquina small.

La máquina m1.large puede detectar y normalizar 500 fotografías en 127 segundos, menos de la mitad del tiempo que necesitaba la máquina small.

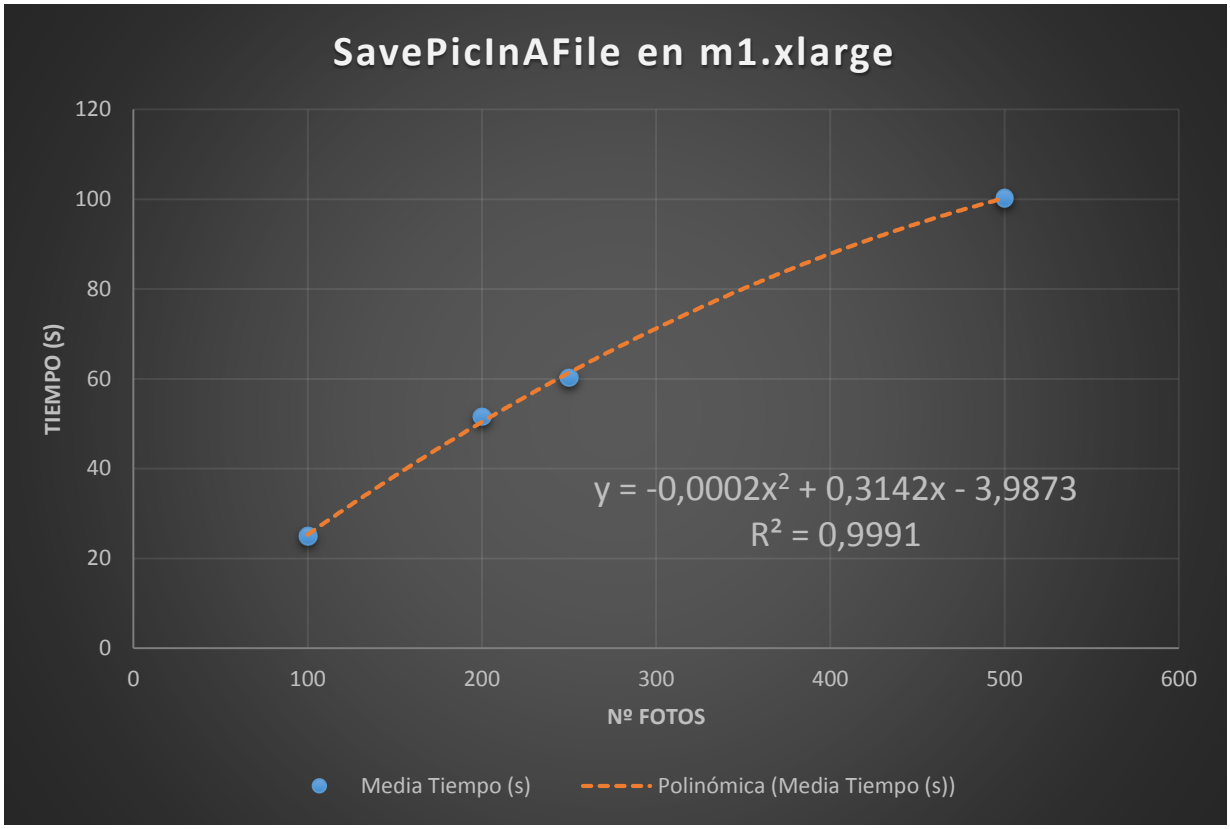

Fig. 6.4 Gráfica SavePicInAFile en m1.xlarge

En el gráfico (Fig. 6.4) podemos apreciar el rendimiento de la máquina m1.xlarge frente al programa SavePicInAFile. Los cuatro procesadores que posee esta máquina le proporciona una gran potencia de computación.

La potencia de esta máquina permite la detección y normalización de 500 fotos en 100 segundos. Podemos observar que multiplicar x2 el número de procesadores no implica reducir el tiempo a la mitad.

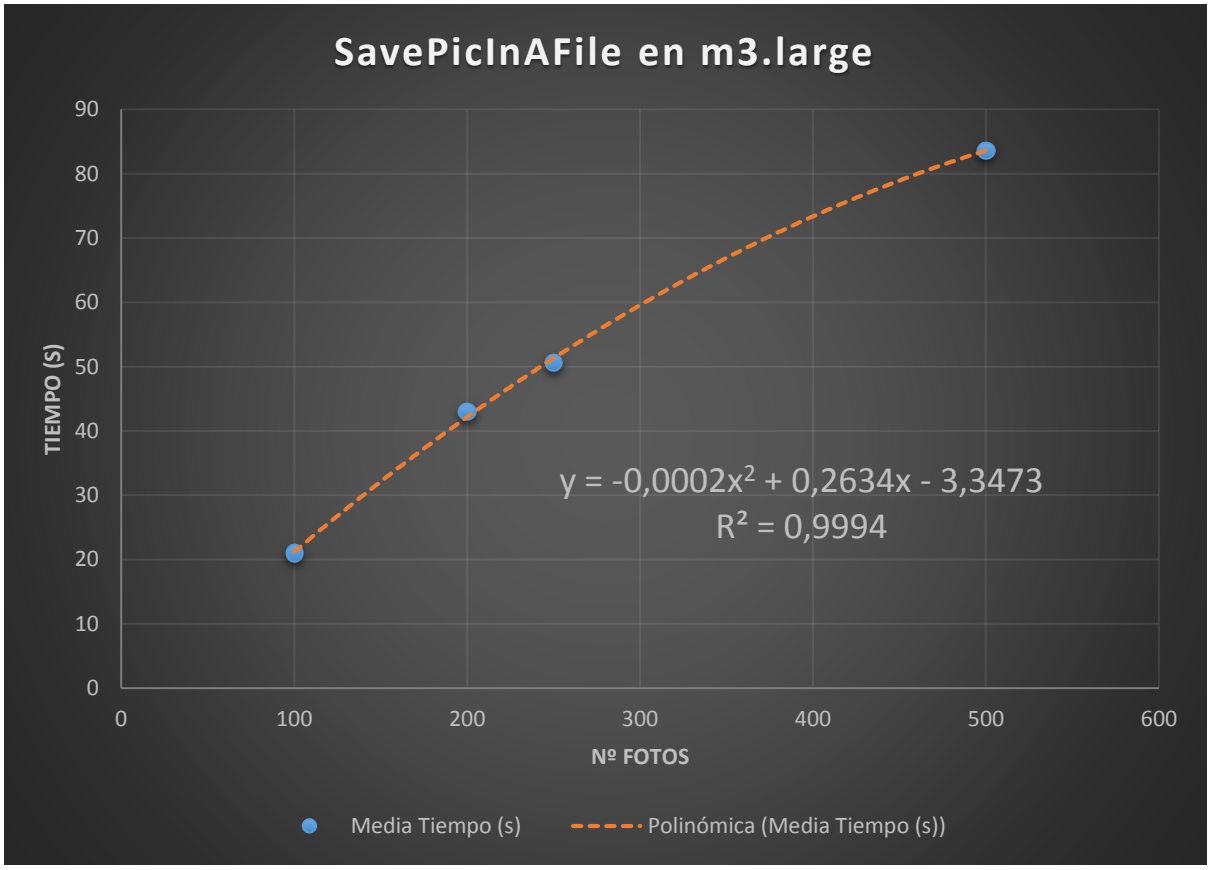

Fig. 6.5 Gráfica SavePicInAFile en m3.large

En la gráfica (Fig. 6.5) podemos ver la mejora introducida por la máquina m3.large. Esta máquina es de la nueva generación y cuanta con el último procesador Xeon y discos duros SSD.

Se puede apreciar una mejora en conjuntos de fotos grandes frente a la máquina m1.xlarge. Aun así en conjunto pequeños de fotos la mejora es muy pequeña.

Podemos decir que la m3.large tiene mejor rendimiento que la m1.xlarge pese a ser una máquina de un nivel inferior gracias a su tecnología.

6.1.1 Discusión

El programa SavePicInAFile no necesita una gran potencia de computación ya que en ninguna de las máquinas utilizadas el tiempo de detección y normalización de un rostro supera el segundo. Sin embargo, es imprescindible la rapidez de este programa en el conjunto del sistema de seguridad ya que si no es capaz de gestionar una gran cantidad de imágenes en un tiempo mínimo puede convertirse en el cuello de botella del sistema debido a que es la puerta de entrada al sistema.

Utilizando las ecuaciones podemos observar un comportamiento polinómico de segundo grado. En la Tabla 6.1 se puede ver un resumen de las características generales del programa SavePicInAFile en las distintas máquinas utilizadas utilizando las ecuaciones obtenidas en con las gráficas.

	Tiempo Medio 1 Foto (s)	Tiempo Total 500 fotos (s)
t1.micro	0.3..1.25 (irregular)	-
m1.small	0.64	320
m1.large	0.254	127
m1.xlarge	0.2	100
m3.large	0.167	84

Tabla 6.1 Resultados SavePicInAFile

Dados los resultados obtenidos y la baja potencia de computación que necesita este programa, se ha decido que este programa será ejecutado desde los propios ordenadores a los que están conectadas las cámaras de seguridad, que pueden ser ordenadores individuales o pertenecer un pequeño cluster de computación que podría compartir potencia de computación cuando fuera necesario. De esta forma, disminuiríamos también la cantidad de información enviada a través de la red ya que se enviarían las fotos normalizadas con un tamaño reducido. Por lo tanto no hablaremos de costes asociados al cloud público en este caso.

6.2 FaceFeatures

El programa FaceFeatures encargado de la extracción de los 66 puntos facial de una imagen normalizada y su posterior almacenamiento ha sido probado en las diferentes instancias de AWS EC2 mencionadas anteriormente. Se han utilizado los resultados para construir unos gráficos de los cuales hemos podido extraer la ecuación del rendimiento de cada máquina.

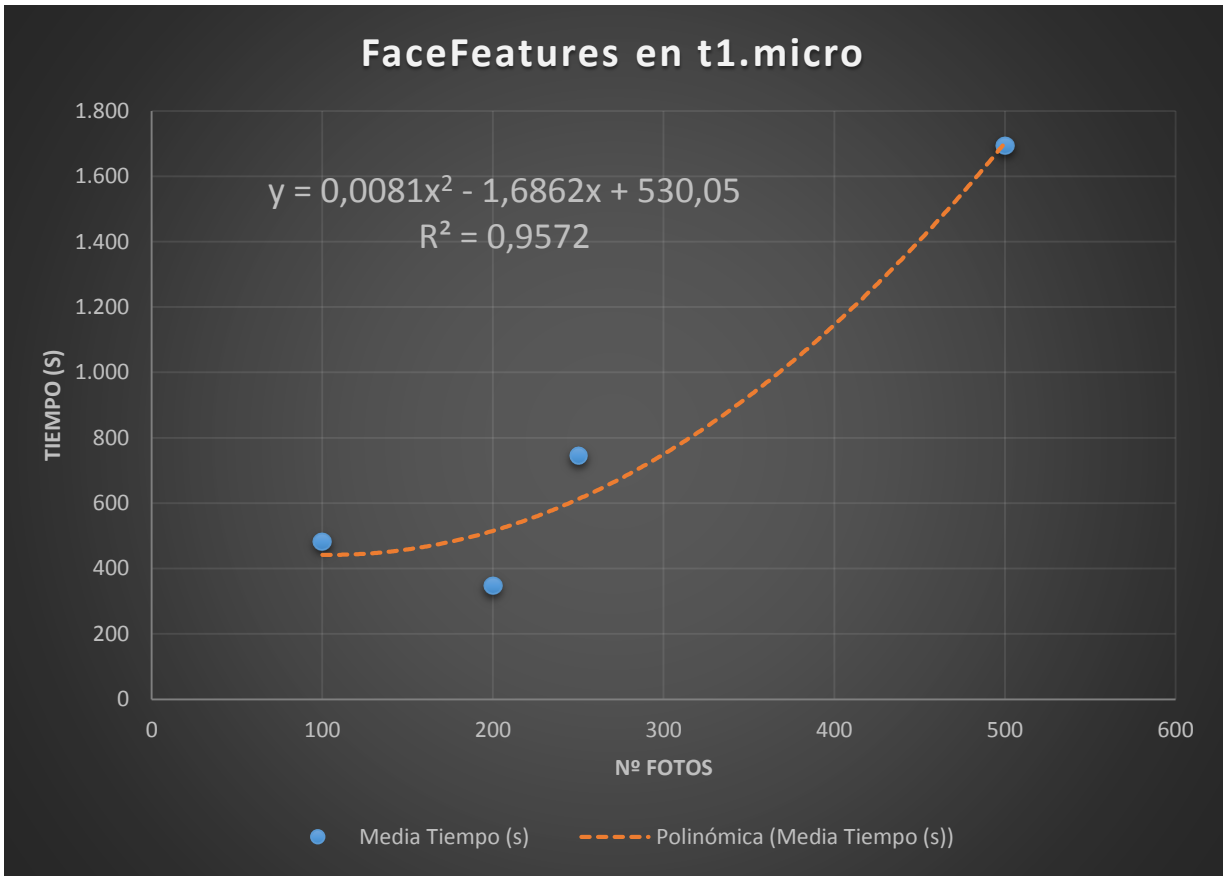

Fig. 6.6 Gráfica FaceFeatures en t1.micro

En el gráfico (Fig. 6.6) se puede ver el rendimiento del programa FaceFeatures en la máquina t1.micro. Al igual que en la prueba realizada en esta instancia con el programa SavePicInAFile, se puede apreciar un comportamiento irregular en las diferentes pruebas. Además, se aprecia un aumento del tiempo de computación respecto al programa SavePicInAFile por lo que podemos concluir que es un programa que necesita más potencia de computación.

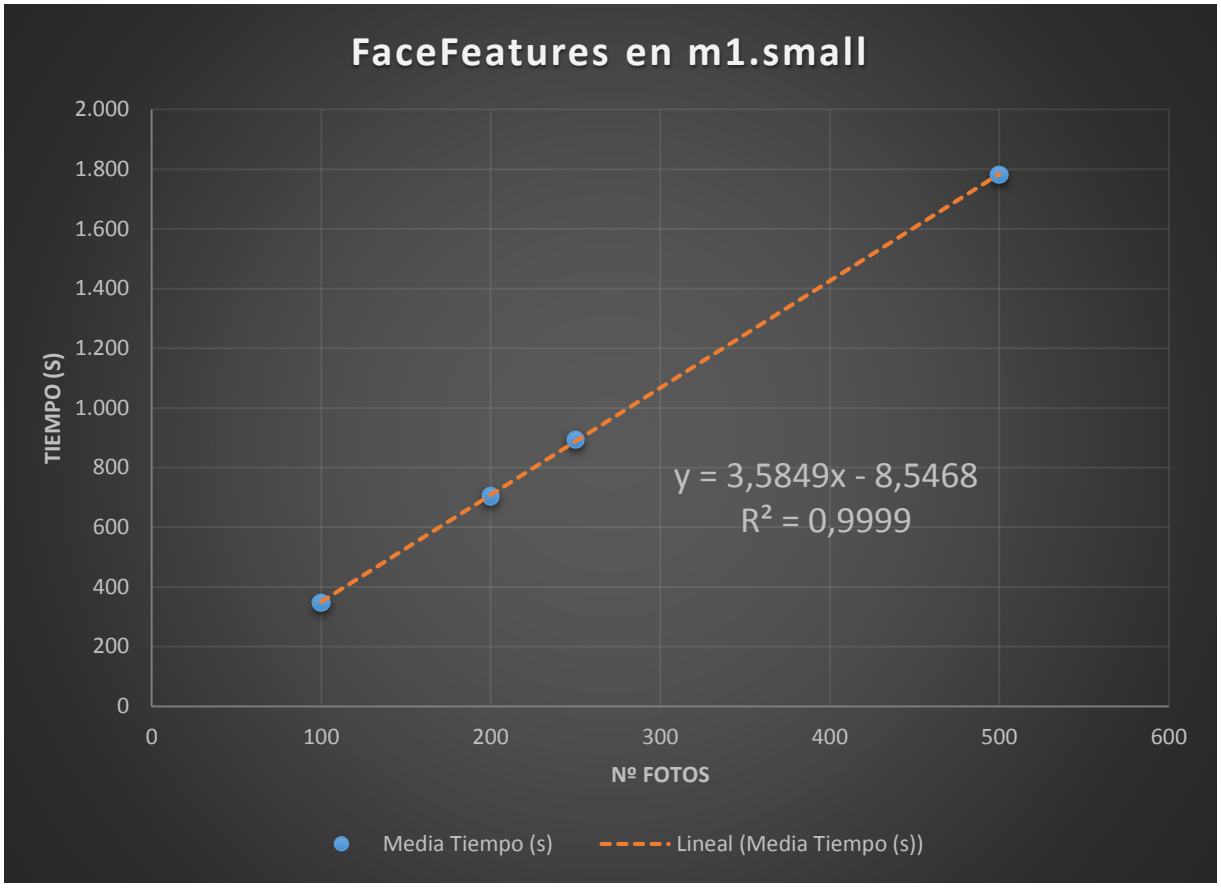

Fig. 6.7 Gráfica FaceFeatures en m1.small

En el gráfico (Fig. 6.7) se puede ver el comportamiento del programa FaceFeatures en la máquina m1.small. Se puede comprobar que, al contrario que la máquina micro, se obtienen unos resultados regulares. En concreto, se puede apreciar un comportamiento lineal del programa lo que permite calcular su rendimiento para otros valores con gran facilidad y fiabilidad.

Aun siendo un comportamiento regular, esta máquina presenta tiempos superiores a los tres segundos por fotografía. Dado que es una fase previa al inicio de la comparación con las bases de datos de fotografías es necesario una mejora de este tiempo de respuesta.

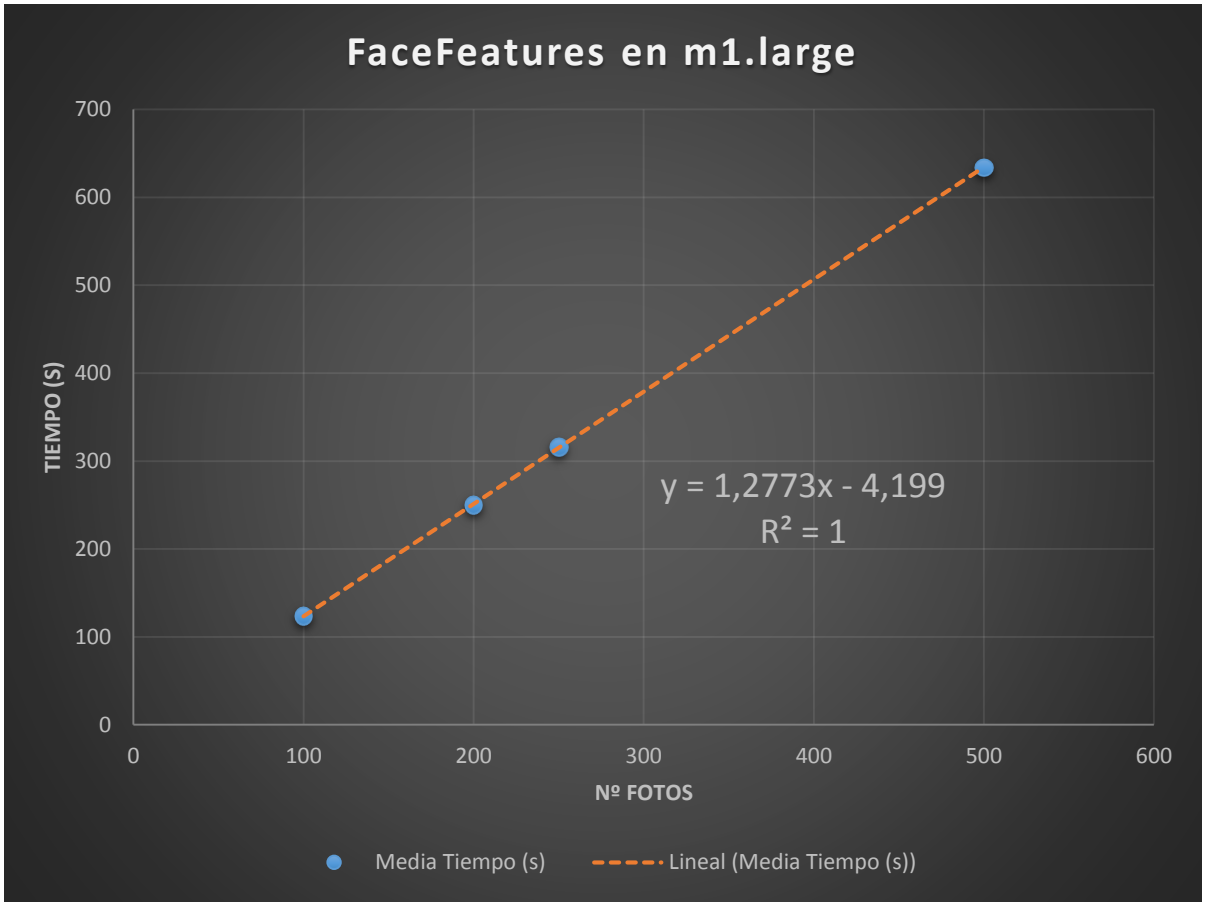

Fig. 6.8 Gráfica FaceFeatures en m1.large

El gráfico (Fig. 6.8) presenta el rendimiento del programa FaceFeatures en la máquina m1.large. Se puede apreciar una mejora del rendimiento respecto a la máquina small, en parte gracias a los dos procesadores posee. Se mantiene el comportamiento lineal en el rendimiento.

Con un tiempo de 1.26 segundos por fotografía de media mejora ampliamente las máquinas anteriores permitiendo un menor retraso en el lanzamiento del programa Match.

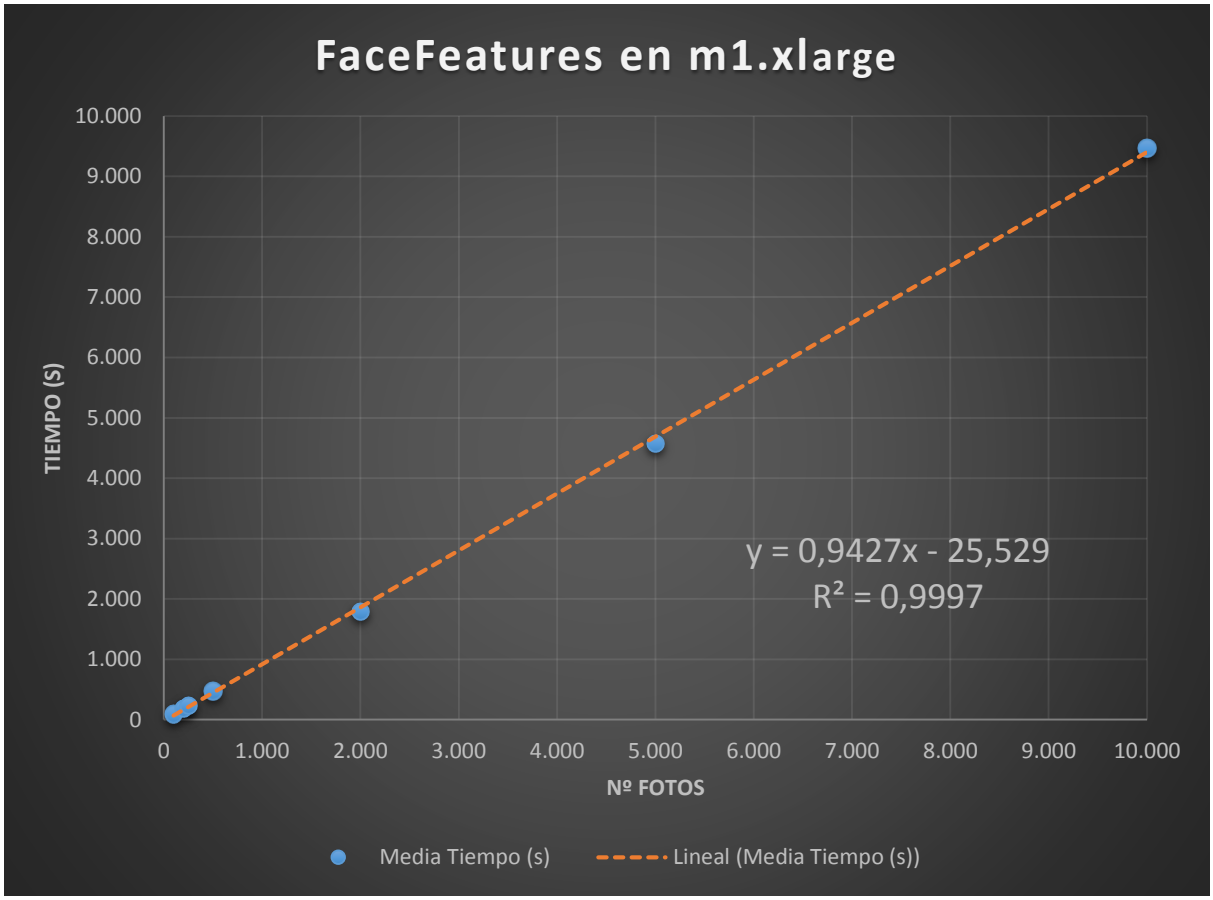

Fig. 6.9 Gráfica FaceFeatures en m1.xlarge

El gráfico (Fig. 6.9) representa el rendimiento del programa FaceFeatures en la máquina m1.xlarge. Manteniendo su comportamiento lineal, el rendimiento en esta máquina ha aumentado considerablemente permitiendo rebajar el tiempo de extracción de los rasgos faciales de una foto a menos de un segundo.

Sin duda alguna, los 4 procesadores que posee esta máquina permite un mejor rendimiento frente a las máquinas anteriores.

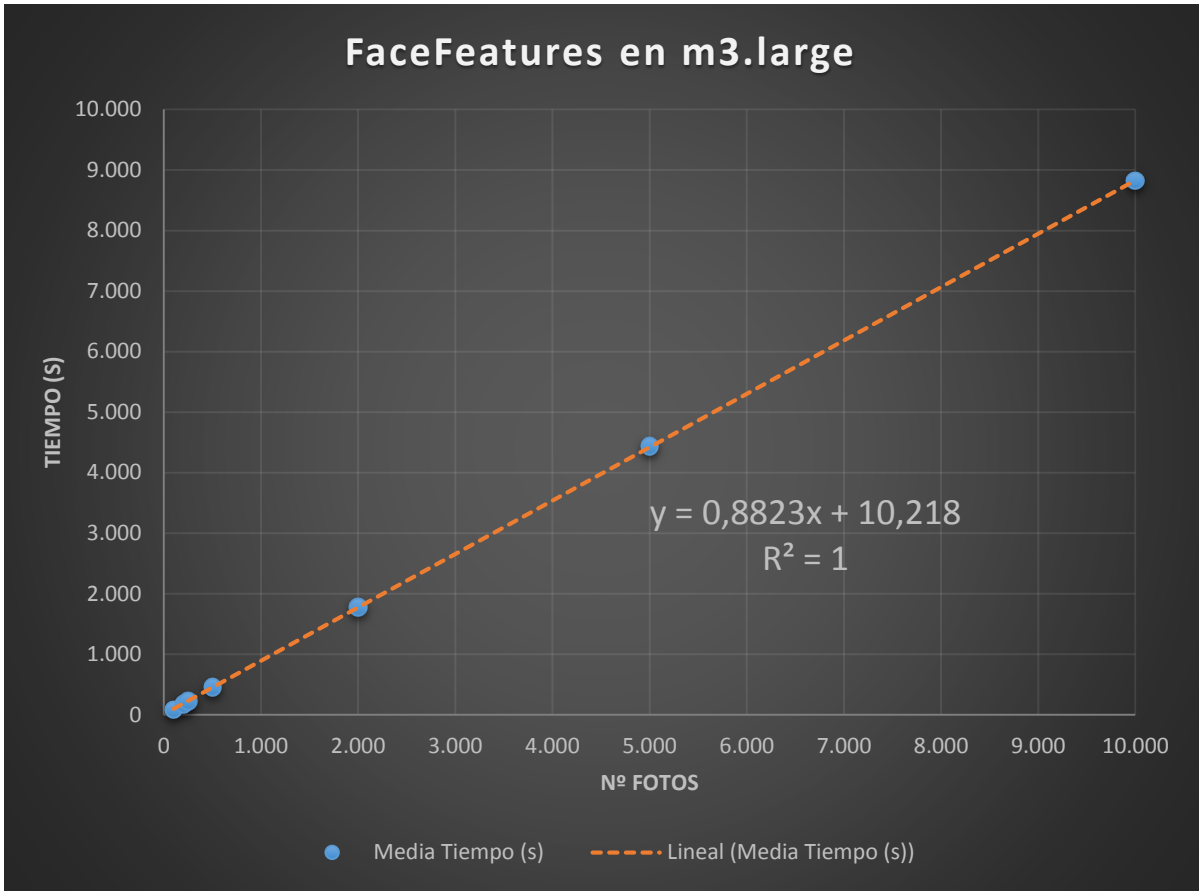

Fig. 6.10 Gráfica FaceFeatures en m3.large

En gráfico (Fig. 6.10) se puede apreciar el rendimiento del programa FaceFeatures en la máquina m3.large. Se puede ver una mejora del rendimiento frente a la m1.xlarge gracias a la mejora en la tecnología de la nueva generación de máquinas. Podemos confirmar el comportamiento lineal que presenta el programa en todas las máquinas.

Con un rendimiento de una extracción de rasgos por cada 0.9 segundos se convierte en la máquina que mejor responde ante este programa.

6.2.1 Discusión

El programa FaceFeatures es la antesala del programa Match. Es imprescindible un buen rendimiento en este programa ya que todas las fotografías que vayan a ser comparadas con las bases de datos tienen que pasar por esta fase obligatoriamente.

El programa FaceFeatures se ejecuta internamente en el programa Match para extraer los rasgos faciales de la fotografía a comparar, por lo tanto el tiempo de extracción estará incluido en los resultados del programa Match. Estos tiempos frente a grandes cantidades de fotografías pueden parecer un problema ya que, si de media el tiempo de extracción de los rasgos es de 1 segundo, el tiempo total del reconocimiento aumentaría en 1 segundo por fotografía. Es no es cierto ya que estos tiempos quedan reducidos gracias a la paralelización en las máquinas ejecutando varias comprobaciones a la vez y al gran número de máquinas utilizadas por el sistema, suponiendo este incremento de tiempo un aumento casi despreciable frente al tiempo de la comparación de las bases de datos.

En la Tabla 6.2 podemos observar las resultados generales del programa FaceFeatures en las distintas máquinas:

	Tiempo Medio 1 Foto (s)	Tiempo Total 500 fotos (s)
t1.micro	1.5…5 (irregular)	-
m1.small	3.56	1782
m1.large	1.26	634
m1.xlarge	0.94	480
m3.large	0.88	458

Tabla 6.2 Resultados FaceFeatures

6.3 Match

El programa Match encargado de la comparación de una fotografía con otras muchas almacenadas en una base de datos para dar un tanto por ciento de similitud, ha sido probado en las diferentes instancias de AWS EC2 mencionadas anteriormente. Se han utilizado los resultados para construir unos gráficos de los cuales hemos podido extraer la ecuación del rendimiento de cada máquina.

6.3.1 Discriminación

Debido a la gran cantidad de máquinas probadas y dentro de estas, las distintas posibilidades de distribución de trabajo en las máquinas con más de una CPU, se ha realizado una discriminación comparando los rendimientos de una máquina con las diferentes distribuciones del trabajo para elegir la mejor posibilidad dependiendo del tamaño de la base de datos de imágenes.

6.3.1.1 Máquina m1.large

Esta máquina posee dos CPU por lo que se han comparado las 2 posibilidades:

- 1 trabajo para las 2 CPU
- 2 trabajos para las 2 CPU

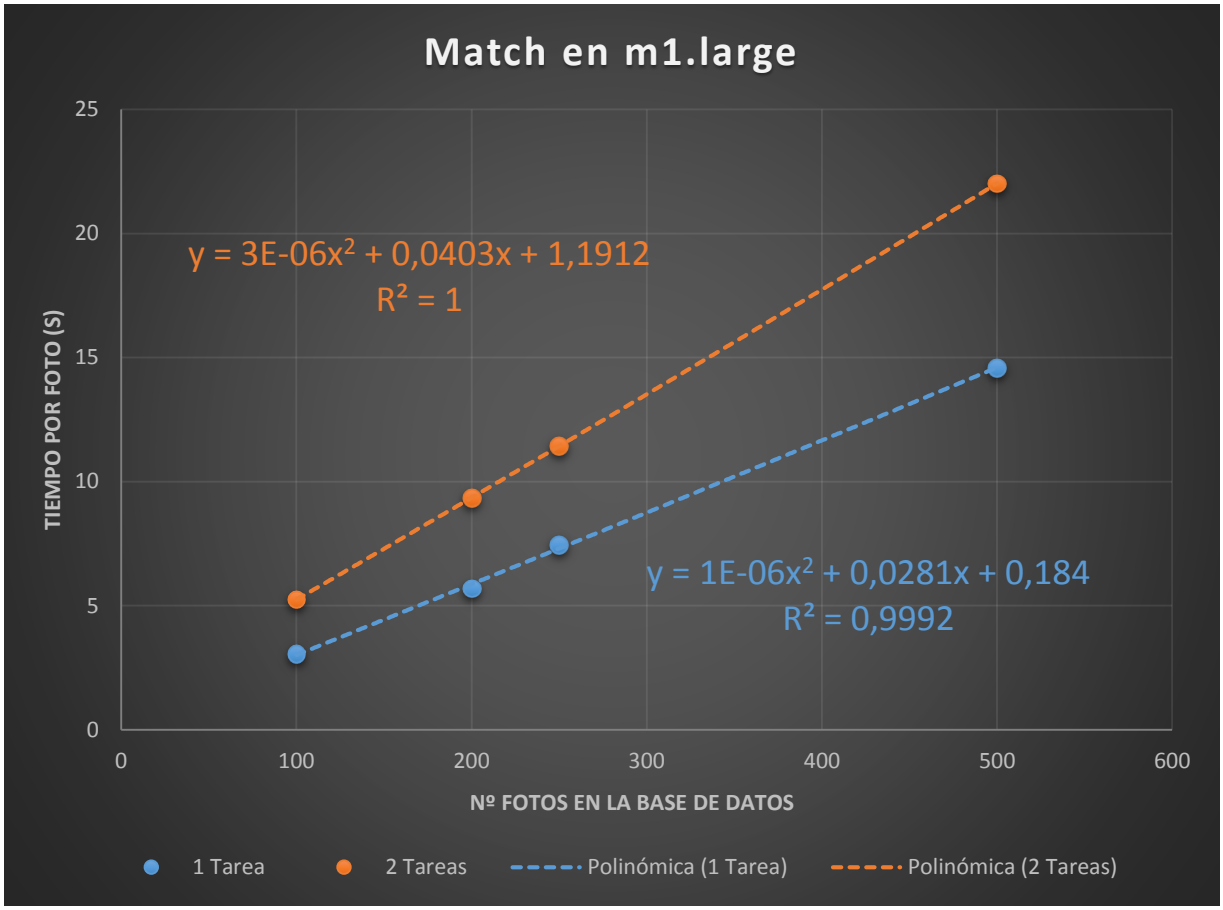

Fig. 6.11 Gráfica Match en m1.large

En el gráfico (Fig. 6.11) se puede ver el rendimiento del programa Match en la máquina m1.large con las 2 posibles distribuciones de trabajo, dándonos el tiempo de comparación de una foto con todas las almacenadas en la base de datos. En el caso de 2 tareas ejecutadas al mismo tiempo, hay que recordar que el número de fotos comparadas es el doble que el número que aparece en el gráfico.

Este gráfico no nos permite saber cuál de las 2 configuraciones es mejor, por ello utilizaremos la inversa de la ecuación obtenida para saber cuántas fotos puede comparar cada configuración en un tiempo en concreto. Para ello, se debe fijar el tamaño de la base de datos, en nuestro utilizaremos una base de datos de 500 fotos y otra de 1000.

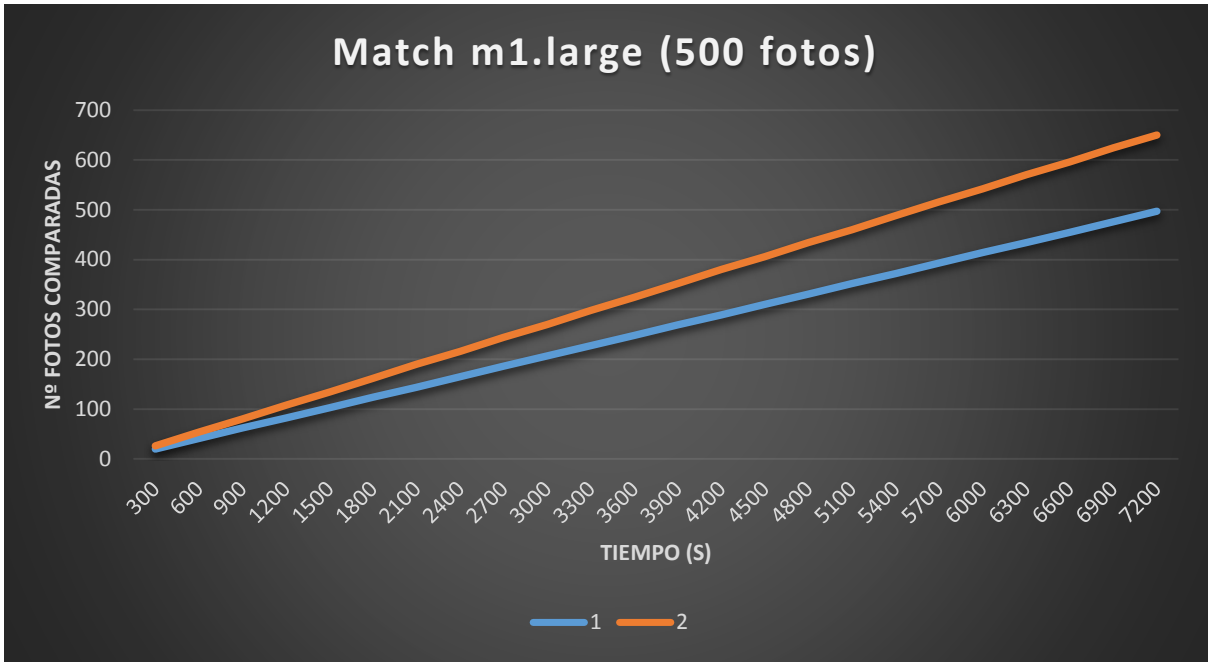

Fig. 6.12 Rendimiento Match m1.large 500 fotos

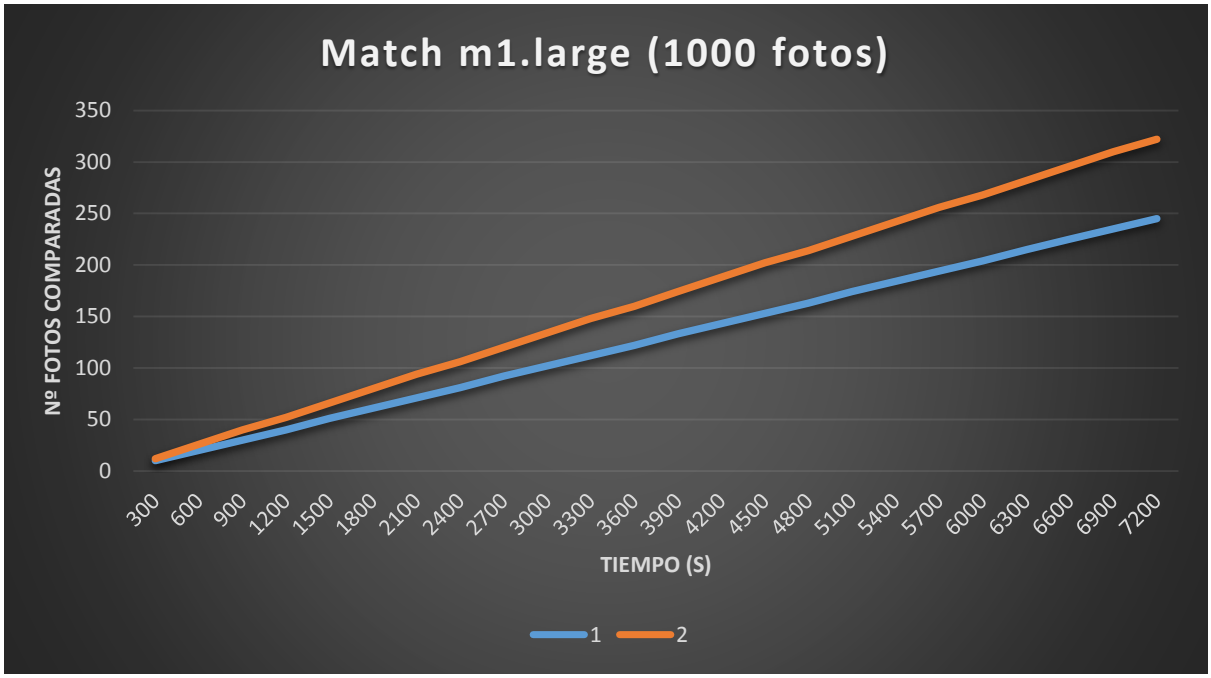

Fig. 6.13 Rendimiento Match m1.large 1000 fotos

Se puede apreciar que independientemente del tamaño de la base de datos, en esta máquina el mejor rendimiento lo proporciona la distribución de trabajo de 2 tareas por máquina logrando resultados superiores a la otra opción de distribución.

Por ello, seleccionaremos esta máquina con esta configuración para bases de datos de tamaño reducido.

6.3.1.2 Máquina m1.xlarge

Esta máquina tiene 4 CPU por lo que se han comparado las 4 posibilidades:

- 1 tarea para las 4 CPU
- 2 tareas para las 4 CPU
- 3 tareas para las 4 CPU
- 4 tareas para las 4 CPU

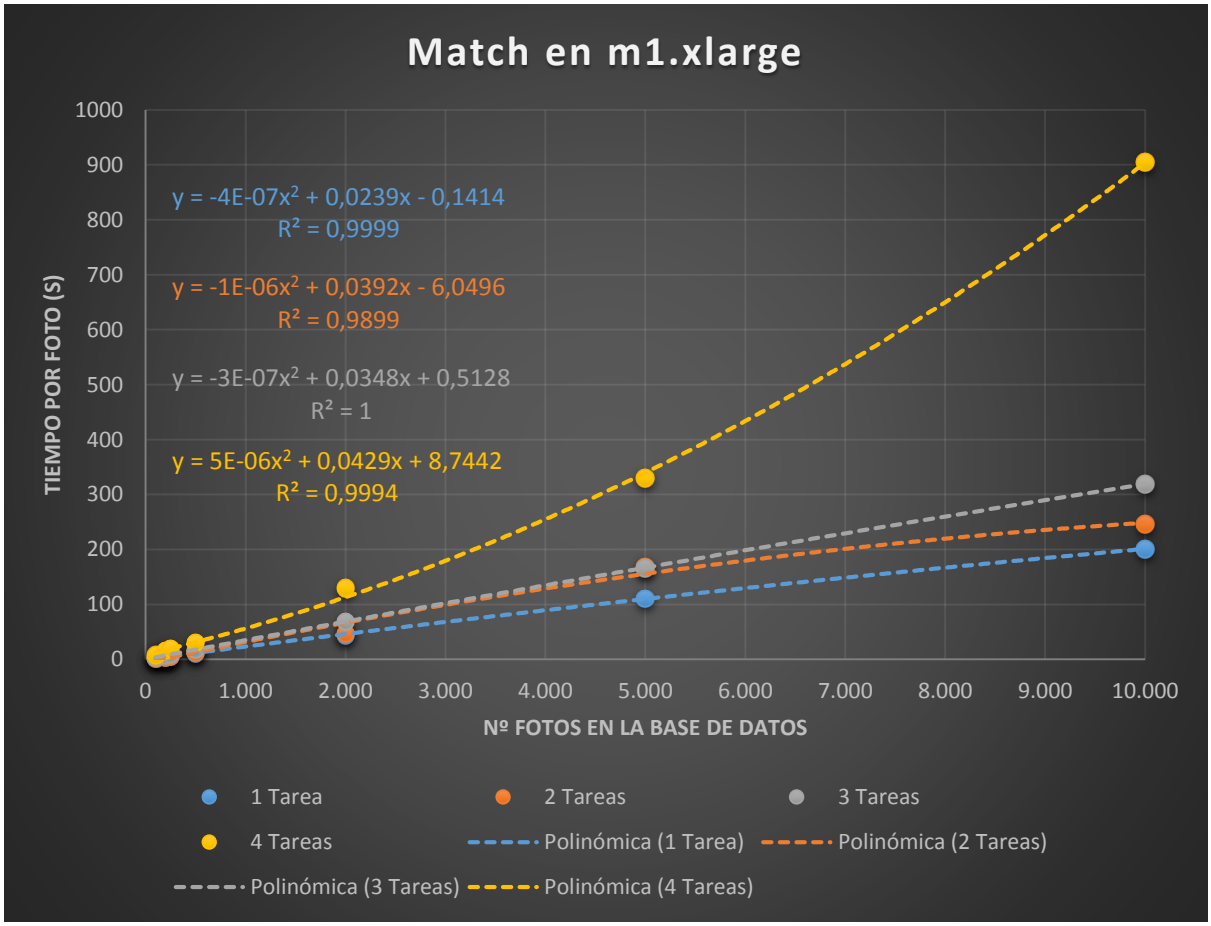

Fig. 6.14 Gráfica Match en m1.xlarge

En el gráfico (Fig. 6.14) se puede ver el rendimiento del programa Match en la máquina m1.xlarge en sus diferentes configuraciones. Mientras que 1,2 y 3 tareas al mismo tiempo dan resultados satisfactorios, el rendimiento de 4 tareas por máquina

se va degradando cada vez más según va aumentando el tamaño de la base de datos. En bases de datos de pequeño tamaño no queda tan claro por ello, realizaremos el mismo gráfico que en la máquina anterior utilizando la inversa de las ecuaciones obtenidas utilizando bases de datos de distintos tamaños.

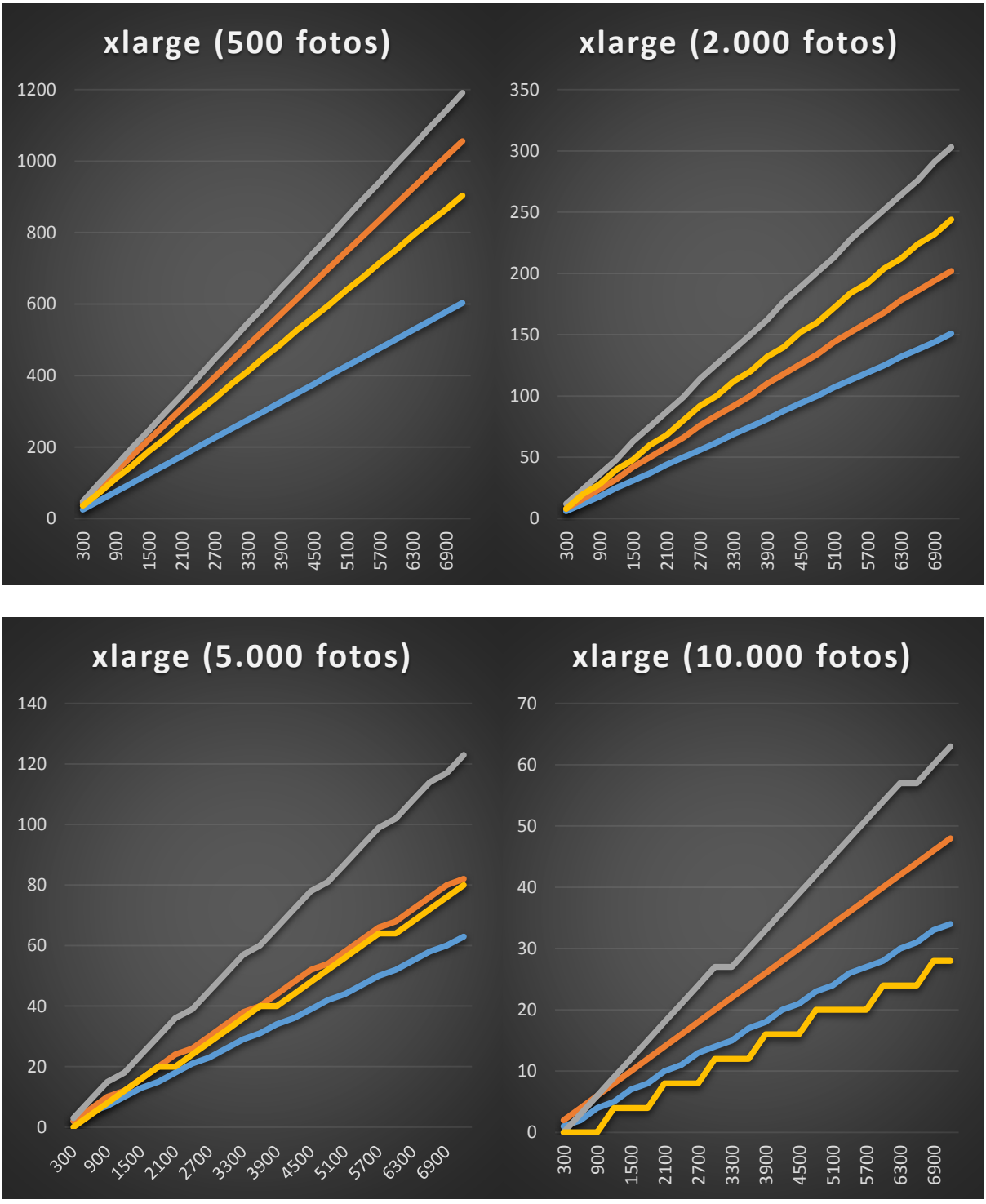

Fig. 6.15, 6.16, 6.17 y 6.18 1 Gráfica Rendimiento m1.xlarge

Leyenda: 1 2 3 4 Eje x: Tiempo (s), Eje y: Nº Fotos comparadas

Después de estudiar los gráficos anteriores, se puede concluir que la mejor configuración para esta máquina es ejecutar 3 tareas al mismo tiempo obteniendo resultados superiores a las otras tres configuraciones independientemente del tamaño de la base de datos dentro de los datos conocidos.

6.3.1.3 Máquina m3.large

Esta máquina perteneciente a la nueva generación de máquinas que provee AWS EC2 posee igual que la m1.xlarge 4 procesadores por lo que posee las mismas configuraciones:

- 1 tarea para las 4 CPU
- 2 tareas para las 4 CPU
- 3 tareas para las 4 CPU
- 4 tareas para las 4 CPU

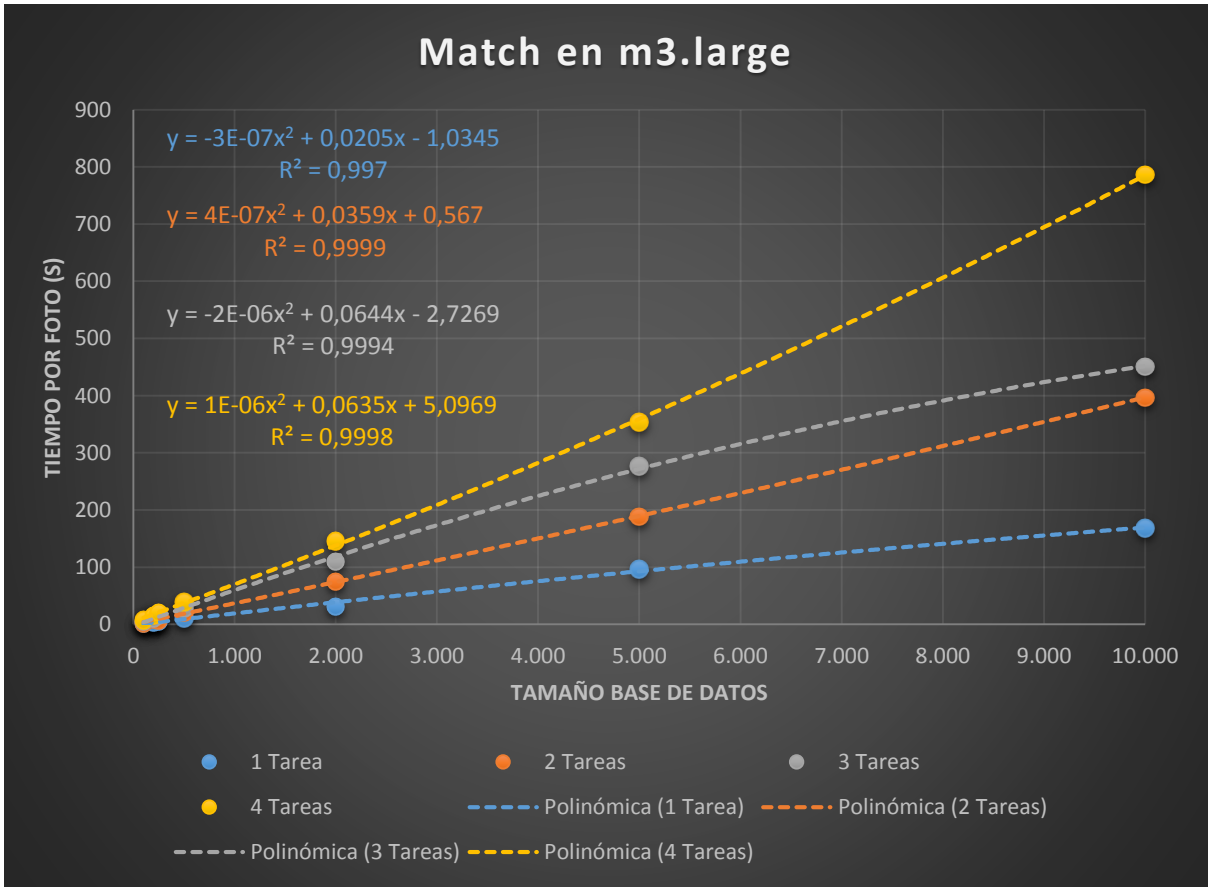

Fig. 6.19 Gráfica Match en m3.large

En el gráfico (Fig. 6.19) se puede ver el rendimiento del programa Match en la máquina m3.large en sus distintas configuraciones. Podemos observar que al igual que en la máquina m1.large, la distribución de 1,2 o 3 tareas en la máquina da resultados satisfactorios mientras que de nuevo, utilizar 4 tareas por máquina degrada su rendimiento.

Utilizando la inversa de las ecuaciones obtenidas de las máquinas se han realizado unos gráficos como en las máquinas anteriores para evaluar el rendimiento real en ciertos intervalos de tiempo.

El gráfico 6.20 presenta prácticamente un empate entre las 4 posibles distribuciones, siendo la distribución 1 tarea para las 4 CPU la más eficiente por muy poco. Esto se debe a la mejora en el procesador y el almacenamiento de estas máquinas. Se muestra un comportamiento similar en los gráficos 6.21 y 6.22 en los que la mejor distribución sigue siendo la anteriormente mencionada.

Sin embargo, para una base de datos de 10.000 fotografías la tendencia cambia y es la distribución de 3 tareas en cada máquina la que mejor rendimiento da, sufriendo la distribución de 4 tareas por máquina un gran decrecimiento en su rendimiento quedando incluso por debajo de otra distribución y empatada con otra.

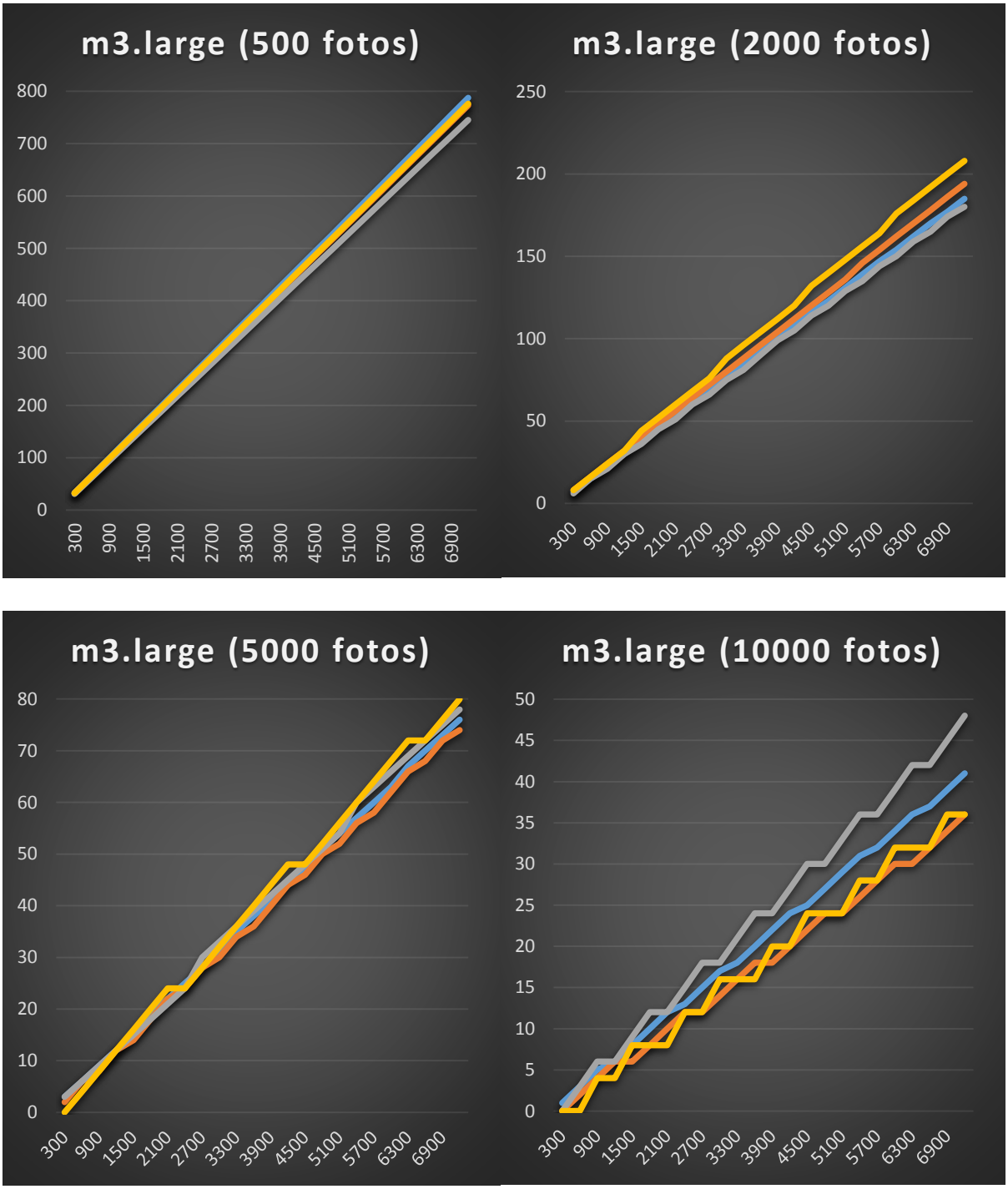

Fig. 6.20, 6.21, 6.22 y 6.23 Rendimiento Match en m3.large

Leyenda:

Eje x: Tiempo (s), Eje y: Nº Fotos comparadas

6.3.1.4 Máquina m1.small y t1.micro

La máquina small debido a su bajo rendimiento y a su única CPU queda muy por debajo de sus competidores en la ejecución del programa Match, por lo que la utilización de esta máquina para este programa solo sería posible si utilizamos bases de datos de tamaño muy reducido.

La máquina micro presenta un rendimiento todavía peor que la máquina small y además la irregularidad de tiempos la excluye del conjunto de máquinas que pueden ejecutar este programa en tiempos concretos.

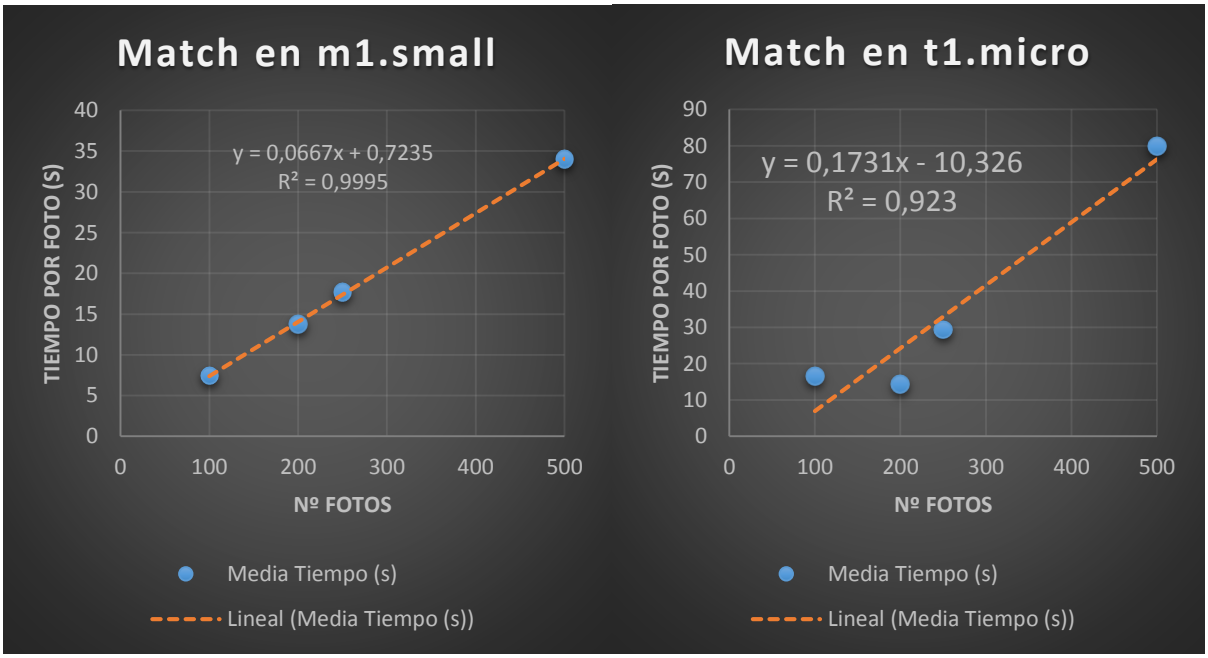

Fig. 6.21 y 6.22 Gráficas Match en m1.small y t1.micro

6.3.2 Flujo de Datos y Costes

En esta sección se van a mostrar los resultados obtenidos tras enfrentar las distintas máquinas seleccionadas en el apartado anterior frente a distintos flujos de datos. Se mostrará el tiempo total para comparar un número concreto de fotografías en cada máquina y el coste que conllevaría utilizarla.

Como se ha dicho anteriormente, AWS EC2 utiliza el sistema de pago de pago-por-uso fraccionado en franjas de tiempo de 1 hora, es decir, el precio por utilizar una máquina 1 hora será independiente del uso que demos a dicha máquina. Por ello, existirán combinaciones más óptimas que nos permitan tener un buen rendimiento con el menor coste posible.

Las máquinas seleccionadas con sus configuraciones correspondientes que mejor rendimiento presentan con sus costes por hora, frente a una base de datos de 10.000 individuos aparecen en la Tabla 6.3:

Máquina	Coste ($ por hora)
m1.xlarge (3 t)	$0.350
m3.large (3 t)	$0.140
m3.large (1 t)	$0.140
m1.large (2 t)	$0.175
m1.small	$0.044

Tabla 6.3 Precios de las máquinas de AWS EC2

A continuación se presentan cuatro gráficos con los distintos rendimientos de las máquinas seleccionadas frente a bases de datos de 500, 2000, 5000 y 10000 fotos. En los gráficos se muestra la cantidad de fotografías comparadas en un tiempo determinado.

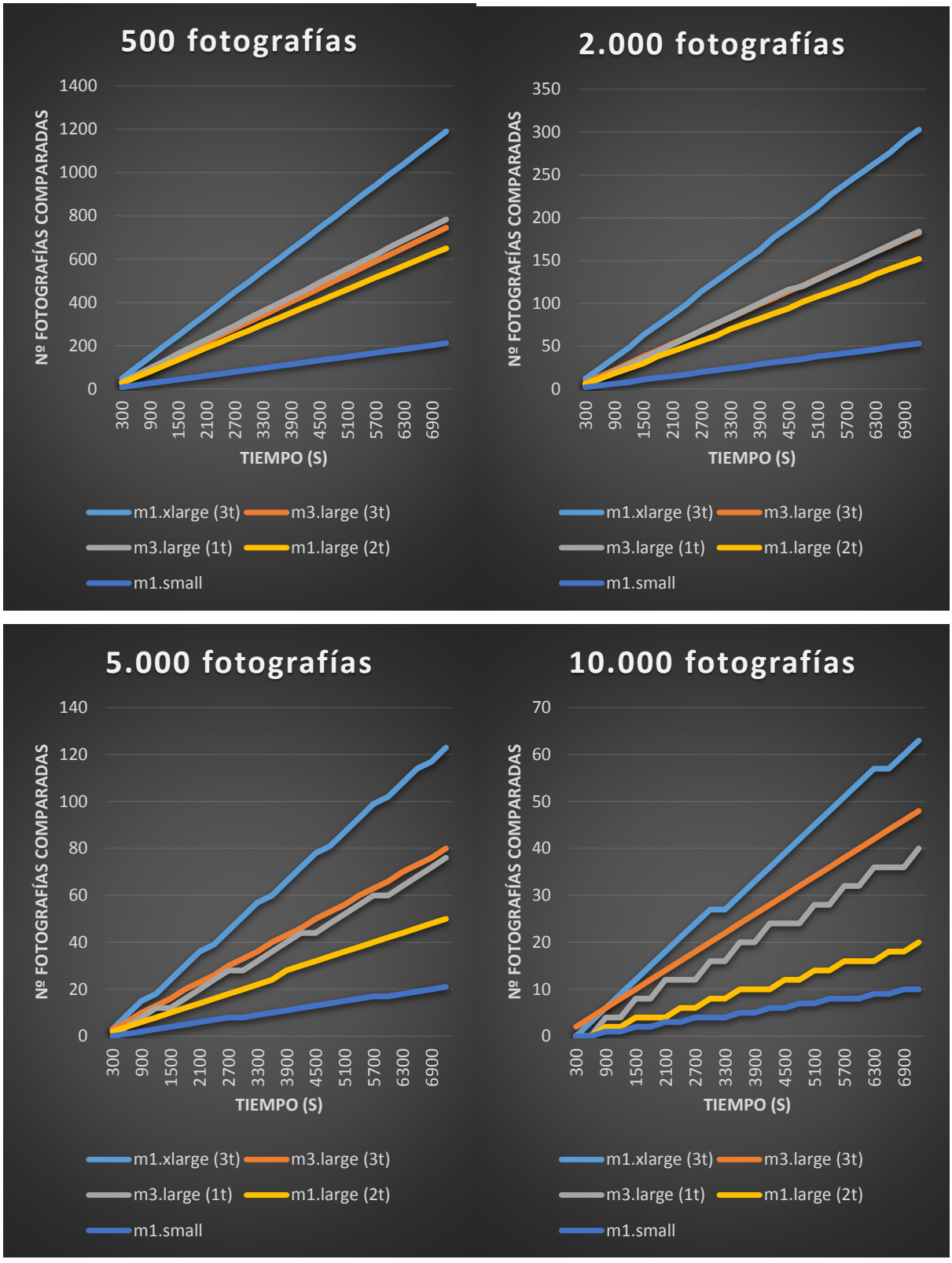

Fig. 6.23, 6.24, 6.25 y 6.26 Rendimiento Match en m3.large

La máquina m1.xlarge bajo la configuración de 3 tareas por máquina, es la que mejor rendimiento proporciona ante cualquier tamaño de base de datos. Sin duda alguna, si lo que queremos es rendimiento es nuestra opción. Sin embargo, si aumentamos el número de máquinas utilizadas y estas tienen un precio inferior es posible obtener un mejor resultado que con la máquina m1.xlarge.

A continuación se muestran unos gráficos en los que podemos ver el tiempo de ejecución de un flujo de individuos concreto y su coste asociado en distintas máquinas. Para ello, se han utilizado los datos de rendimiento de las máquinas y el coste que tienen por hora pero antes hay que hacer unas aclaraciones:

- **Tiempo mínimo de ejecución**: Este tiempo será el tiempo de análisis de una foto en una determinada máquina y es independiente del número de máquinas que se utilicen o del tamaño de la base de datos. No se puede ejecutar el programa en menos tiempo.

- **Número máximo de instancias**: Este número está íntimamente relacionada con el tiempo mínimo de ejecución. Indica el número máximo de instancias que se puede utilizar para ejecutar un determinado flujo de datos en el tiempo mínimo de ejecución. Este número lo podemos obtener aproximadamente con un margen de error muy pequeño con la siguiente fórmula:

  Nº Máx. Máquinas = Tamaño flujo de datos / Nº de tareas por máquina

- **El coste en AWS EC2** está fraccionado por horas. Una vez pagada la hora de una máquina su precio será el mismo se utilice o no.

Los resultados mostrados en los siguientes gráficos han sido obtenidos bajo las siguientes características: la base de datos de imágenes se fija a 10.000 fotografías y un flujo de individuos de 1.000 personas por hora.

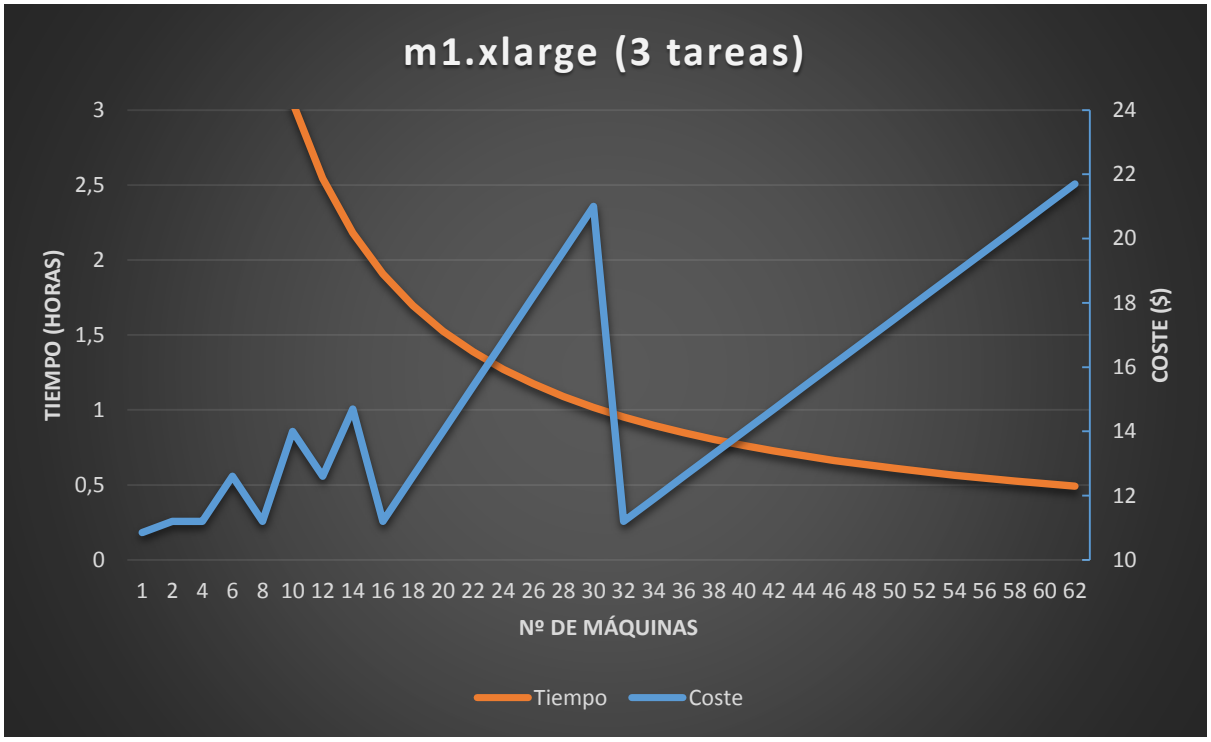

Fig. 6.30 Coste/Tiempo Match en m1.large 3T

En el gráfico (Fig. 6.30) podemos observar el rendimiento de la máquina m1.xlarge frente a un flujo de datos de 1.000 individuos. Para poder procesar esos 1.000 individuos en 1 hora, son necesarias 32 máquinas con un coste total por hora de 11,2$. Además podemos extraer también la cantidad de máquinas y el coste necesario para hacer frente al doble de individuos. Para procesar 2.000 individuos en 1 hora son necesarias 62 máquinas con un coste total de 21,7$.

Podemos observar que el precio de procesar 1000 fotos en 1 hora o en 2 es el mismo, 11,2$ encontrando de esta forma un punto de ejecución óptimo para este flujo de datos.

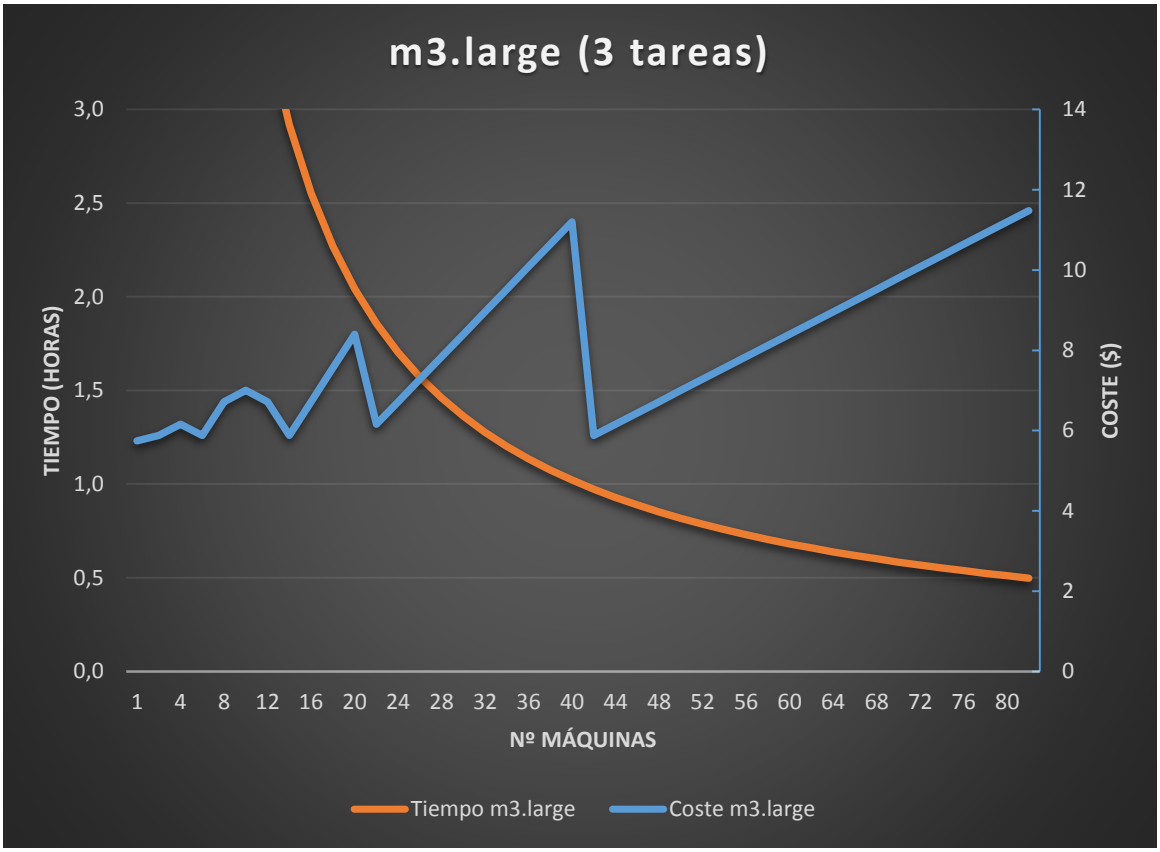

Fig. 6.31 Coste/Tiempo Match en m3.large 3T

En el gráfico (Fig. 6.31) podemos observar el rendimiento de la máquina m3.large (3 tareas) frente a un flujo de datos de 1.000 individuos. Para poder procesar esos 1.000 individuos en 1 hora, son necesarias 42 máquinas con un coste total por hora de 5.88$. Además podemos extraer también la cantidad de máquinas y el coste necesario para hacer frente al doble de individuos. Para procesar 2.000 individuos en 1 hora son necesarias 82 máquinas con un coste total de 11,42$.

Se puede apreciar que pese a que la máquina x1.large había dado un mejor rendimiento en las pruebas anteriores, el precio reducido de la máquina m3.large y su buen rendimiento arroja los mismos resultados con un coste menor.

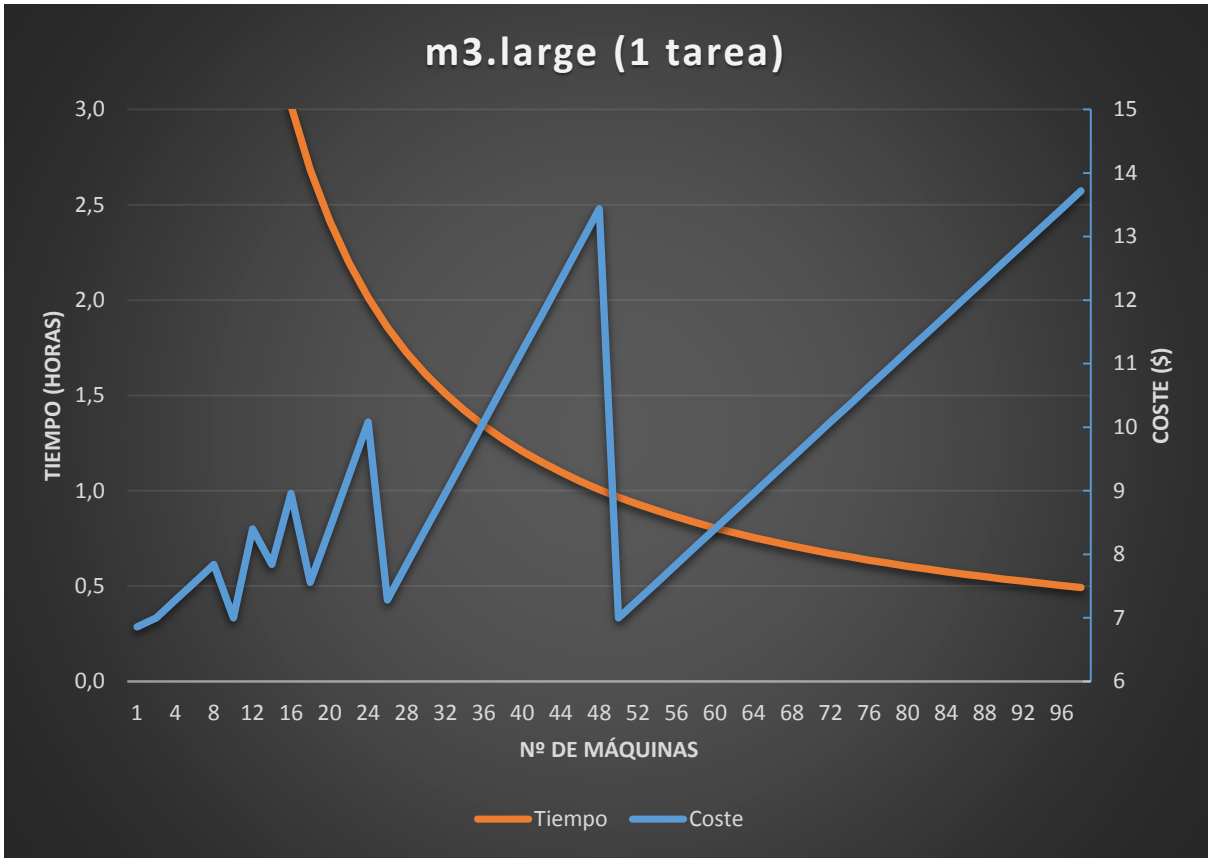

Fig. 6.32 Coste/Tiempo Match en m3.large 1T

En el gráfico (Fig. 6.32) podemos observar el rendimiento de la máquina m3.large (1 tareas) frente a un flujo de datos de 1.000 individuos. Para poder procesar esos 1.000 individuos en 1 hora, son necesarias 50 máquinas con un coste total por hora de 7$. Además podemos extraer también la cantidad de máquinas y el coste necesario para hacer frente al doble de individuos. Para procesar 2.000 individuos en 1 hora son necesarias 98 máquinas con un coste total de 13,2$.

A pesar de no ser de las que mejor rendimiento tiene frente a bases de datos de 10.000 fotografías, la máquina m3.large (1 tareas) obtiene unos resultados superiores a la m1.xlarge respecto al coste, utilizando una mayor cantidad de máquinas. Sin embargo, la m3.large (3 tareas) tiene el mismo rendimiento y mejor coste con menos máquinas.

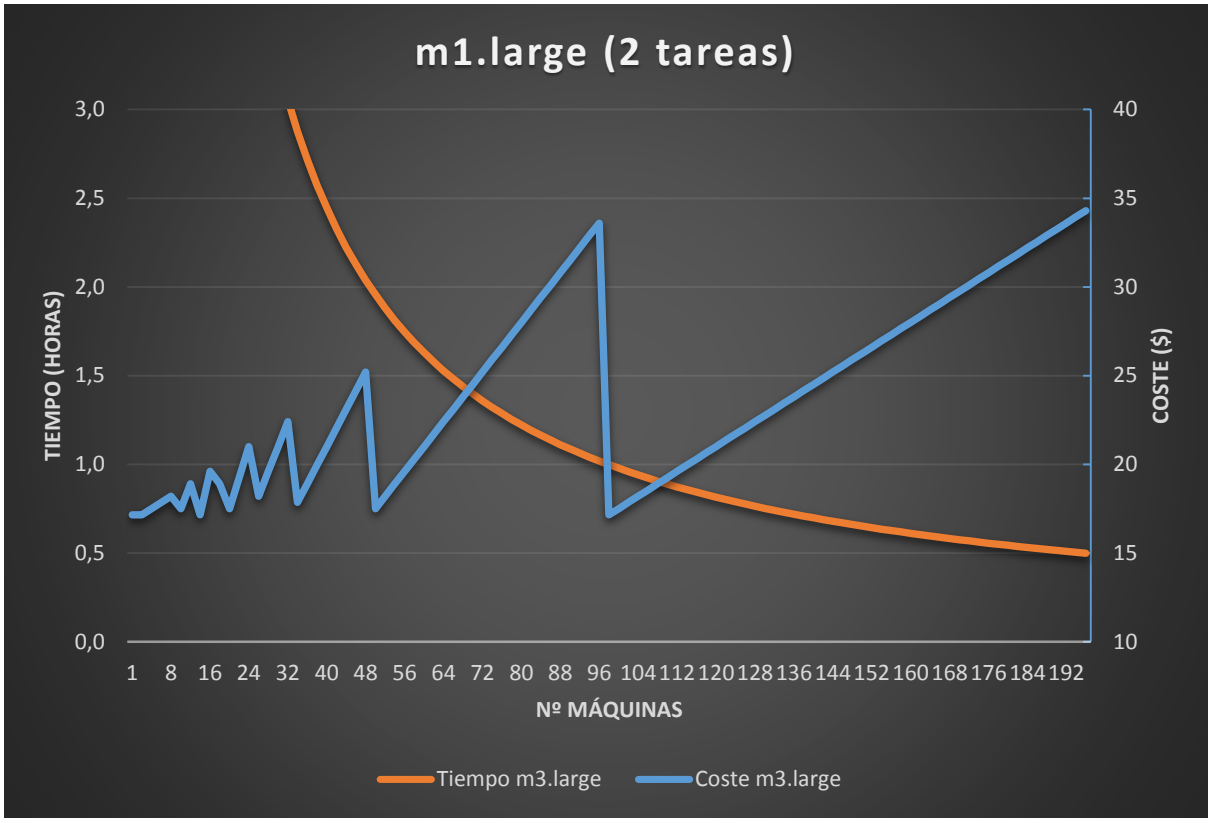

Fig. 6.33 Coste/Tiempo Match en m1.large 2T

En el gráfico (Fig. 6.33) podemos observar el rendimiento de la máquina m1.large (2 tareas) frente a un flujo de datos de 1.000 individuos. Para poder procesar esos 1.000 individuos en 1 hora, son necesarias 98 máquinas con un coste total por hora de 17.15$. Además podemos extraer también la cantidad de máquinas y el coste necesario para hacer frente al doble de individuos. Para procesar 2.000 individuos en 1 hora son necesarias 196 máquinas con un coste total de 34,3$.

Debido a la menor potencia de esta máquina, solamente dos procesadores y su mayor precio que la m3.large hace que esta máquina pierda interés frente a este tipo de bases de datos de gran tamaño. El rendimiento es inferior respecto a sus competidores y el coste aumenta considerablemente.

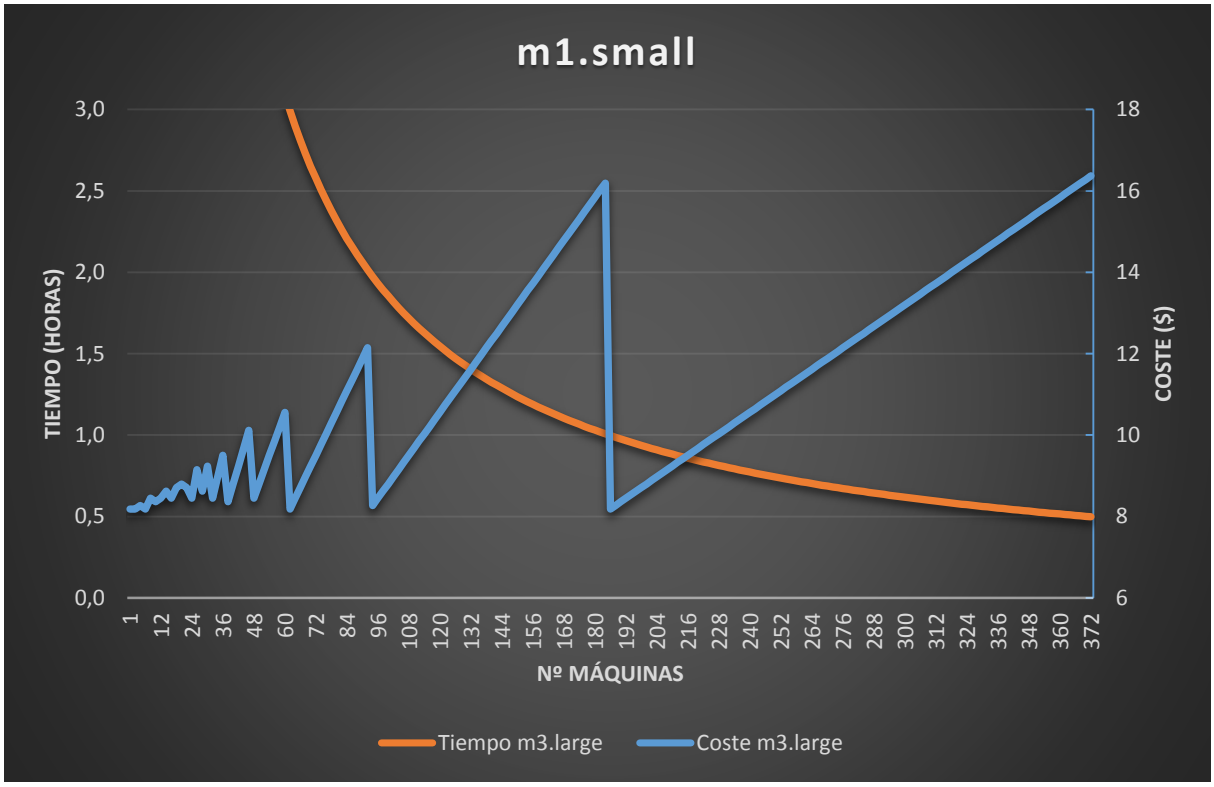

Fig. 6.34 Coste/Tiempo Match en m1.small

En el gráfico (Fig. 6.34) podemos observar el rendimiento de la máquina m1.small frente a un flujo de datos de 1.000 individuos. Para poder procesar esos 1.000 individuos en 1 hora, son necesarias 186 máquinas con un coste total por hora de 8,184$. Además podemos extraer también la cantidad de máquinas y el coste necesario para hacer frente al doble de individuos. Para procesar 2.000 individuos en 1 hora son necesarias 372 máquinas con un coste total de 16,368$.

Podemos concluir que si se puede hacer uso de una gran cantidad de máquinas, la máquina small ofrece un gran rendimiento con un coste muy reducido compitiendo de cerca con la m3.large.

6.3.3. Discusión

El programa Match es la parte más importante del sistema de reconocimiento facial. En él se gastan la mayoría del tiempo y de los recursos de computación ya que muchas veces es necesario acabar el análisis de las fotos en un tiempo límite.

En el análisis y comparación de las fotos uno de los factores más importantes es el tamaño de la base de datos ya que a mayor tamaño mayor tiempo para realizar la comparación. Las pruebas realizadas con las distintas máquinas de AWS EC2 con una base de datos de 10.000 individuos y un flujo de 1.000 personas por hora han arrojado los siguientes resultados:

	Nº Máquinas	Coste ($)	Nº Máx. Máquinas	Tiempo Mín. Ejec. (s)
M1.xlarge (3 tareas)	32	$11.2	333	318.96s
M3.large (3 tareas)	42	$5.88	333	450.89s
M3.large (1 tareas)	52	$7.28	1000	168.95s
M1.large (2 tareas)	98	$17.15	500	704.09s
M1.small	186	$8.184	1000	667.72s

Tabla 6.4 Resultados Match

Si nos centramos en el coste, el menor coste nos lo da la máquina m3.large (3 tareas) costando 42 instancias de esta máquina $5.88 por hora. Sin embargo, su tiempo mínimo de ejecución por foto es de 450 segundos (7 minutos y 30 segundos) por lo que no es útil para sistemas que quieran trabajar cercanos al tiempo real.

Si lo que nos interesa es un tiempo de ejecución mínimo muy bajo para sistemas de seguridad críticos con poco tiempo para el reconocimiento la máquina adecuada es la m3.large (1 tarea) con un precio de $7.28 la hora y un tiempo mín. de 168 segundos (2 minutos y 48 segundos).

La máquina m1.xlarge pese a obtener los mejores resultados de rendimiento, su alto coste y su tiempo de ejecución mínimo de 318 segundos deja a esta máquina en una situación complicada frente a sus competidores de la generación m3.

Por último la máquina m1.xlarge queda totalmente descargada para este tamaño de bases de datos debido a su alto precio comparado con la generación m3 y sus altos tiempos de ejecución por foto. La máquina m1.small obtiene un coste muy reducido de $8.184 por hora, sin embargo, la gran cantidad de instancias que se necesitan para manejar tal información y sus altos tiempos de ejecución mín. hacen que no pueda competir con la m3.large.

Como conclusión, dependiendo de cuál sea la prioridad, si el coste o el tiempo límite que tiene el sistema para analizar una foto elegiremos unas máquinas u otras. Al ser un problema de Big Data dada la gran información que se analiza en poca cantidad de tiempo las máquinas más útiles y con mejores resultados son las más potentes. La nueva generación m3 responde mucho mejor al programa obteniendo resultados que dan viabilidad al sistema de seguridad presentado en este trabajo.

Capítulo 7

7. Caso de Uso: Aeropuerto de Barajas Adolfo Suarez

El aeropuerto de Barajas, es uno de los aeropuertos más grandes e importantes de España. Cada año pasan por allí entre 35 y 40 millones de pasajeros durante las distintas épocas del año siendo los momentos de más afluencia las vacaciones de verano y las navidades en invierno. El aeropuerto está dividido en 4 terminales: T1, T2, T3, T4 y la nueva T4S.

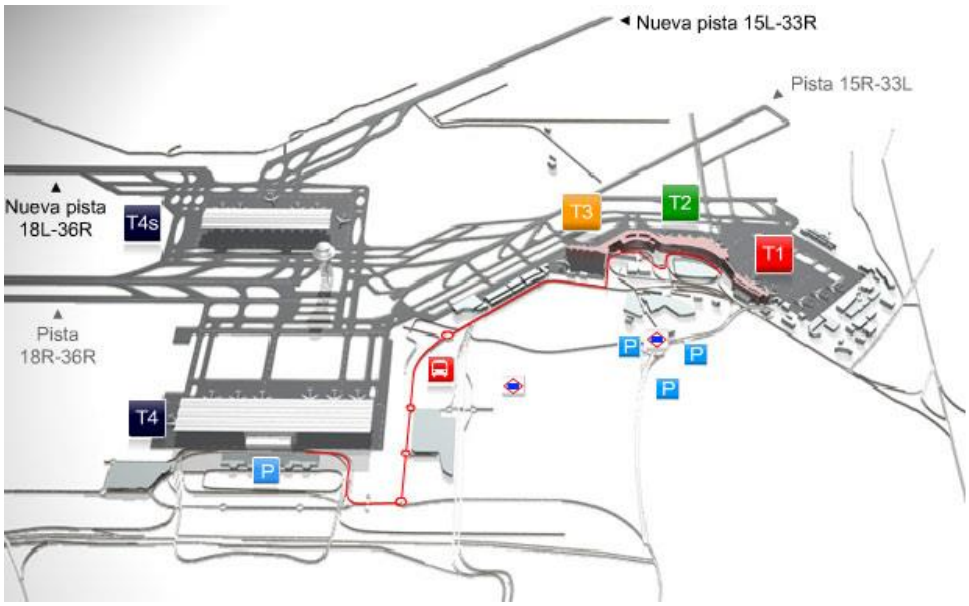

Fig. 7.1 Distribución del aeropuerto de Barajas

En el año 2013 el aeropuerto recibió 39.729.027 pasajeros a lo largo del año siendo el aeropuerto más transitado de España [57], es decir, al día el aeropuerto recibió aproximadamente 108.846 pasajeros.

Si tenemos en cuenta que los últimos vuelos salen o entran del aeropuerto a la 1:30 y los primeros que entran y salen lo hacen sobre las 5:30 de la mañana, existen 20 horas en las que el aeropuerto está lleno de pasajeros. Conociendo este dato, el número de pasajeros por hora durante esas horas es de 6047 de media. Con este escenario es posible implantar nuestro sistema de reconocimiento facial cloud en sus dos versiones, tanto la estática como la dinámica.

Para los escenarios estáticos, es necesario fijar el tiempo límite de ejecución, es decir, el tiempo que va a tardar un viajero en abandonar el aeropuerto. Para los pasajeros que van a tomar un avión este tiempo puede ser más alto ya que normalmente los viajeros toman la precaución de llegar antes al aeropuerto. Sin embargo, los pasajeros que llegan en un vuelo pasan menos tiempo en las instalaciones: recoger el equipaje y dependiendo del origen del vuelo un control de seguridad.

En el caso de los escenarios dinámicos, este tiempo puede ser mayor, sin embargo, cuanto mayor sea menor rendimiento obtendremos del sistema. Vamos a fijar los siguientes tiempos límites de ejecución:

- Escenarios estáticos:
    - Pasajeros que van a tomar un vuelo: 8 minutos máx.
    - Pasajeros que llegan de un vuelo: 5 minutos máx.

Dentro de las máquinas estudiadas en este trabajo, las que tienen un tiempo de ejecución mín. menor que 8 minutos son la m1.xlarge (3 tareas) y la m3.large con (1 y 3 tareas). Conociendo el comportamiento de estas máquinas y sus costes asociados se utilizará la máquina m3.large en sus 2 versiones y se descartará la m1.xlarge. Sin embargo, solamente la m3.large (1 tarea) tiene un tiempo mín. de ejecución inferior a 5 minutos.

A continuación se pueden ver los gráficos de la máquina m3.large (1 tarea) y (3 tareas) frente a una base de datos de 10.000 personas. Este número de individuos ha sido obtenido de la página web de la Interpol [58], dónde el número de personas buscadas en Europa junto con América asciende aproximadamente a 10.000.

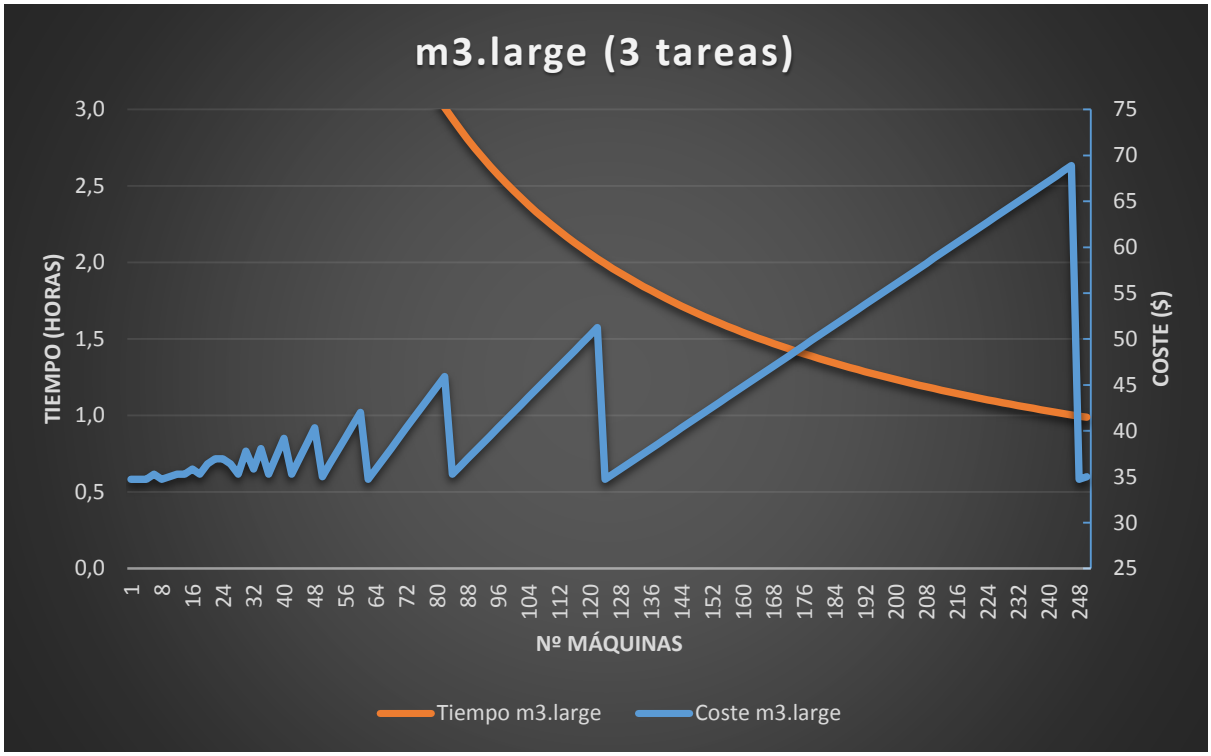

Fig. 7.2 Coste/Tiempo para escenarios estáticos en m3.large 3T

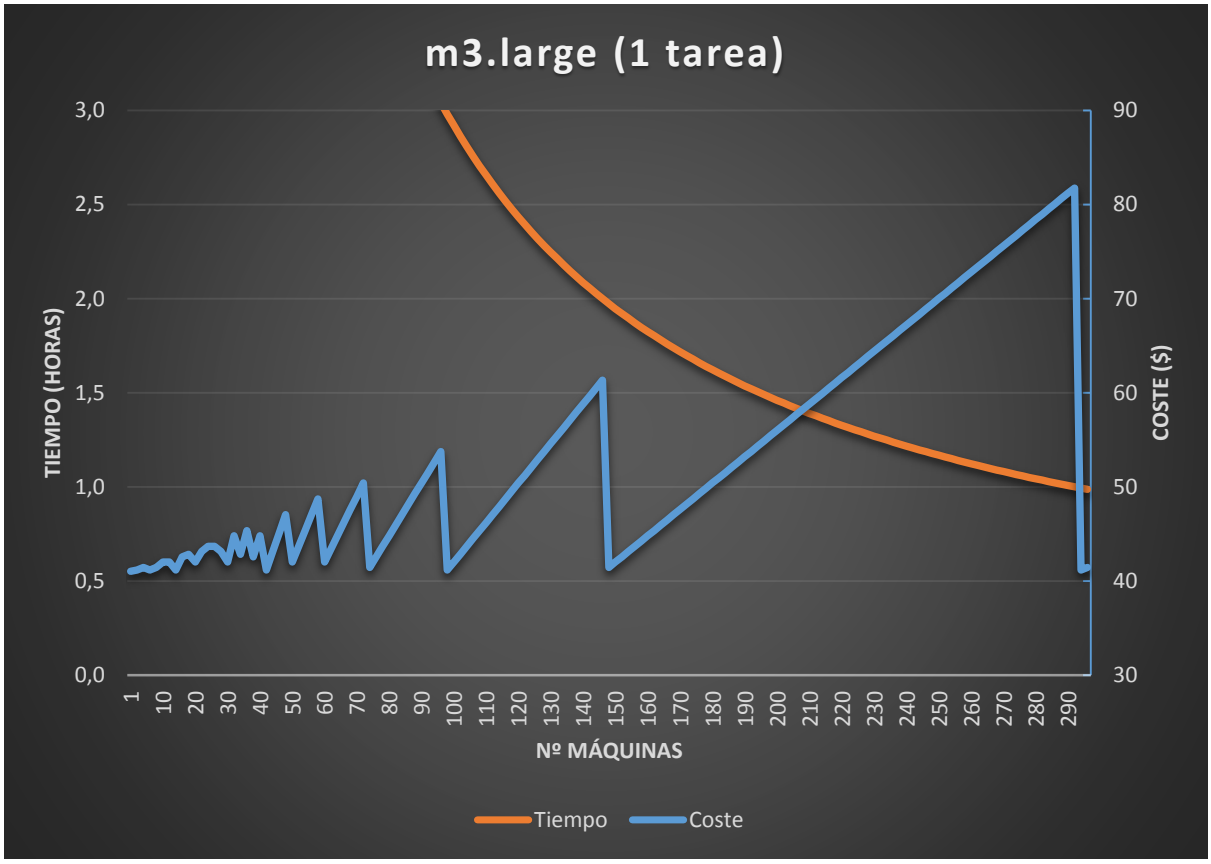

Fig. 7.3 Coste/Tiempo para escenarios estáticos en m3.large 1T

Los datos obtenidos se pueden ver en la Tabla 7.1. Podemos apreciar que el número máquinas que necesitamos de ambas máquinas es bastante alto, sin embargo, eso no es un problema. Se pueden realizar acuerdos con AWS para poder disponer de tal número de máquinas sin ningún problema. Podemos ver como el mejor rendimiento nos lo da la máquina m3.large (1 tarea) con un coste de $41.44 a la hora. Por otro lado, la máquina m3.large (3 tareas) nos da un rendimiento más reducido pero a cambio de una reducción en el coste. Si utilizamos estas máquinas a lo largo del día obtenemos unos costes de $828 y $700 dólares respectivamente.

	Nº Máquinas	Coste/hora ($)	Coste diario (20 h)	Tiempo Mín. Ejec. (s)
M3.large (3 tareas)	250	$35	$700	450.89s
M3.large (1 tarea)	296	$41.44	$828.8	168.95s

Tabla 7.1 Resultados del escenario estático

En el caso de los escenarios dinámicos vamos a suponer que existen alrededor de unas 100 cámaras distribuidas por todo el aeropuerto colocadas estratégicamente. Esas cámaras toman cada tiempo mínimo de ejecución de la máquina elegida una fotografía y detecta a todos los individuos presentes. De media cada cámara detectará unos 15 rostros por lo que en total serán 1500 rostros a analizar. Se marcará un tiempo límite de ejecución de 20 minutos, mucho más alto que en los escenarios estáticos debido a su menor fiabilidad y al contemplar un conjunto más amplio de individuos al mismo tiempo.

A continuación se pueden ver los gráficos de las dos máquinas anteriores preparadas para afrontar el análisis de 1500 rostros cada 20 minutos. Se ha utilizado la misma base de datos que para los escenarios estáticos, 10.000 individuos.

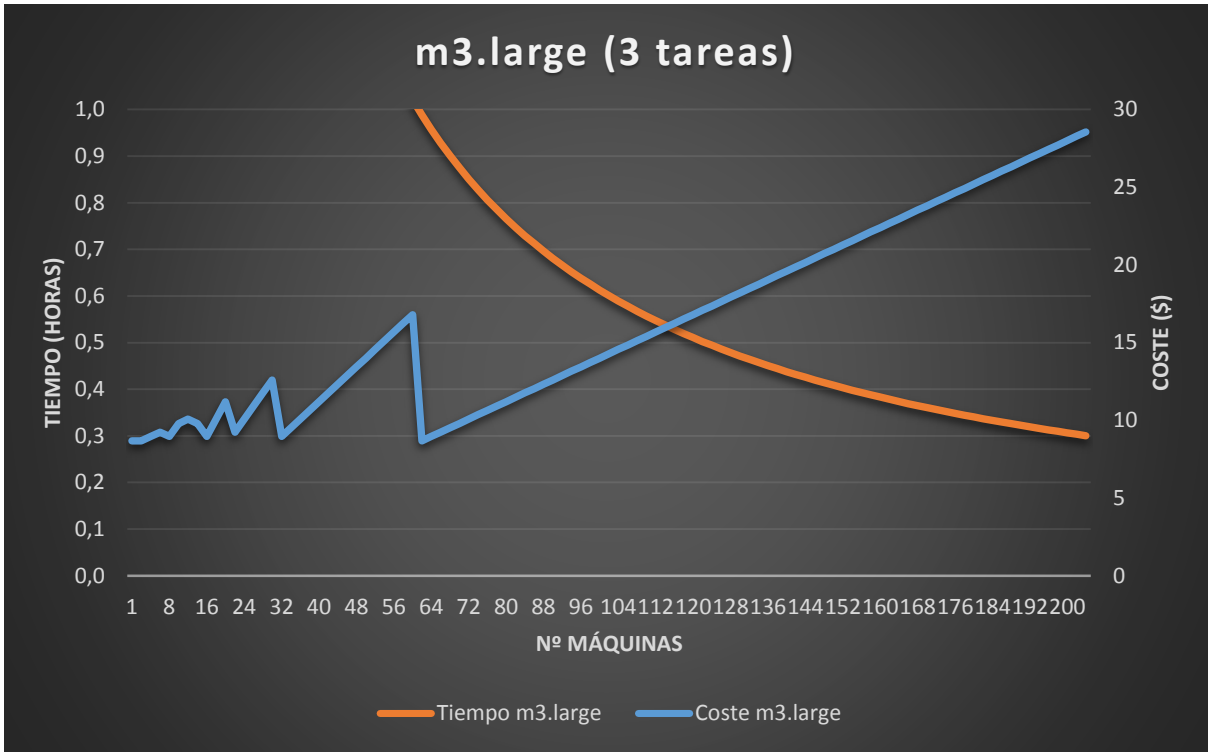

Fig. 7.4 Coste/Tiempo para escenarios dinámicos en m3.large 3T

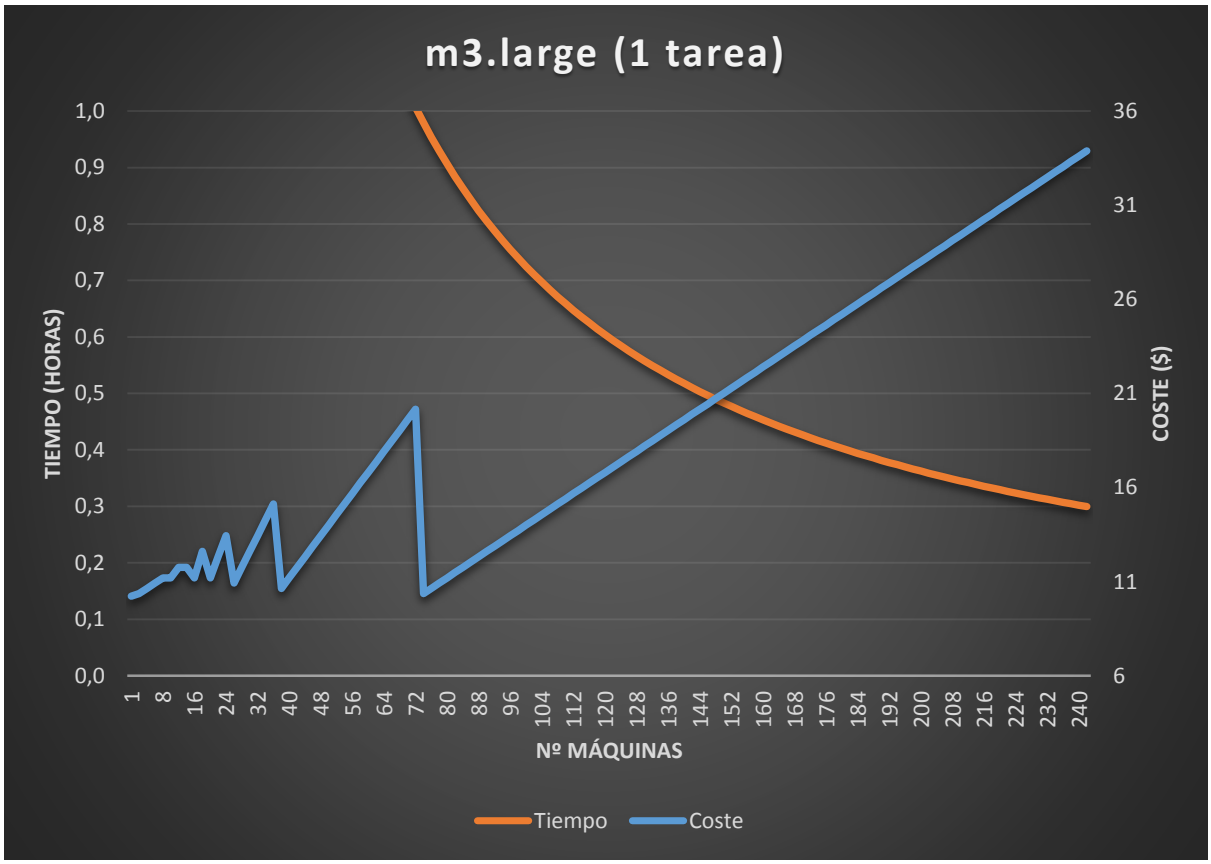

Fig. 7.5 Coste/Tiempo para escenarios dinámicos en m3.large 1T

Se pueden ver los resultados en la Tabla 7.2. Podemos apreciar el alto número de máquinas que necesitamos en este caso también a pesar de relajar el tiempo límite de ejecución. Esto se debe a que, mientras que en modelo estático se necesita asumir un determinado flujo por hora, en el caso del dinámico se debe asumir un flujo menor de individuos pero en 20 minutos. Si realizamos los cálculos apropiados, el sistema es capaz de realizar 4500 comparaciones por hora.

El rendimiento de la máquina m3.large (3 tareas) presenta un coste de $28.56 mientras que la m3.large (1 tarea) uno de $33.88. Dado que ambas tienen un tiempo mín. de ejecución inferior al tiempo límite de ejecución, la opción más óptima es la de menor coste, la m3.large (3 tareas), con un coste diario de $571.2.

	Nº Máquinas	Coste/hora ($)	Coste diario (20h)	Fotos Analizadas por hora	Tiempo Mín. Ejec. (s)
M3.large (3 tareas)	204	$28.56	$571.2	4500	450.89s
M3.large (1 tarea)	242	$33.88	$677.6	4500	168.95s

Tabla 7.2 Resultados escenario dinámico

El precio combinado de estos dos sistemas, el estático con la m3.large (1 tarea) y el dinámico con la m3.large (3 tareas), asciende a $1400 diarios. Existe la posibilidad de establecer acuerdos con AWS para asegurar la disponibilidad de tal cantidad de máquinas. Se pueden realizar reservas anuales, de 3 años o contratos personalizados con AWS de los cuales se obtienen, a parte de la disponibilidad, descuentos en el precio de las máquinas. La máquina m3.large pasaría de $0.140 hora a $0.101. Otra forma de abaratar el precio sería utilizar el mercado de instancias reservadas, en el que se establecería un precio de pago y solamente cuando el precio de esa máquina sea inferior a nuestro precio se utilizarían las máquinas.

Capítulo 8

8. Modelo

Acorde con los resultados y las conclusiones posteriores, se puede obtener un árbol de decisión de estructura dependiendo de nuestras preferencias (Fig. 8.1). Este modelo muestra un árbol de decisión cuyos pasos son los siguientes: el tamaño de la base de datos, el tipo de arquitectura, el tipo de instancia y por último la distribución de trabajo en cada instancia. Para los escenarios estáticos los tiempos de ejecución mínimos superiores a 8 minutos no son aceptados para preservar la calidad del sistema.

El flujo del árbol de decisión de infraestructura pasa por:

1. El proceso de decisión comienza determinando el tamaño de la base de datos que va contener las fotografías y templates de los individuos buscados, que será utilizada por las diferentes arquitecturas.

2. Una vez que el tamaño de la base de datos está fijado, el usuario necesita decidir qué es más importante, mayores condiciones de seguridad o un mayor dinamismo del sistema. Cuando esta decisión es tomada, la arquitectura del Sistema de Reconocimiento Facial Cloud es fijada, eligiendo entre SRF estático o SRF dinámico.

3. Después de fijar el tamaño de la base de datos y la arquitectura, el tipo de instancias es fijado basándonos en el coste, el rendimiento o ambos.

4. La distribución de tareas es automáticamente fijada tras los tres pasos previos, fijando la distribución de tareas que obtiene mejor rendimiento y coste para cada escenario.

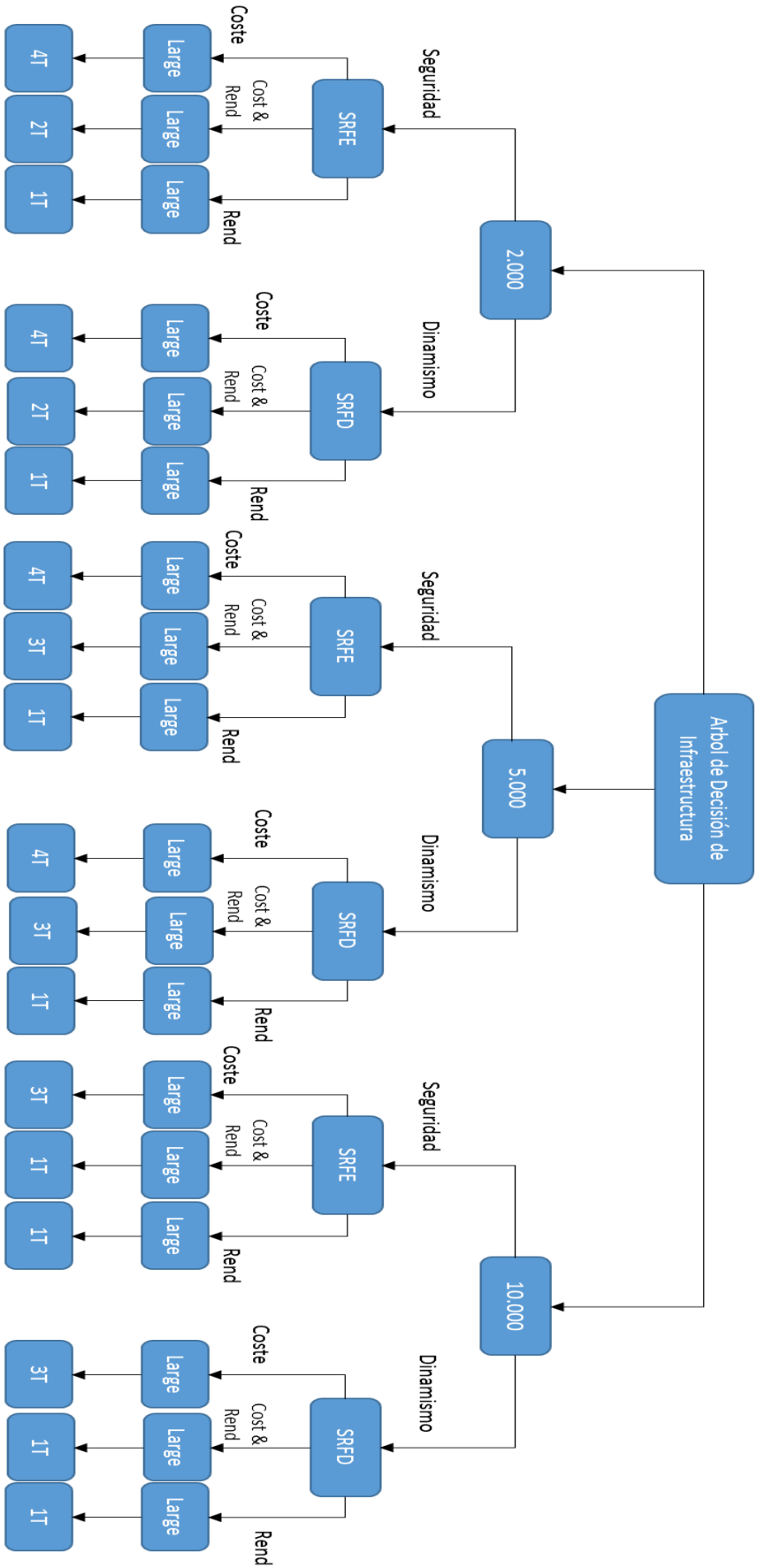

Fig. 8.1 Árbol de Decisión de Infraestructura

Con este modelo, un usuario que quiere desplegar un sistema de seguridad basado en el reconocimiento facial a través del cloud puede decidir facialmente cual es la mejor opción para el contestando unas pocas preguntas básicas para determinar el tipo de máquina que necesita, su distribución y la arquitecturas que concuerda con sus necesidades.

PARTE III

Conclusiones y Trabajo Futuro

Capítulo 9

9. Contribuciones

En este capítulo se habla sobre cuáles son las principales contribuciones del proyecto y se destacan las principales conclusiones sobre el trabajo llevado a cabo.

En concordancia con los resultados obtenidos al analizar los tres programas utilizados en el sistema de seguridad con reconocimiento facial, las dos primeras partes, la detección y la extracción de los rasgos, no necesitan gran potencia de computación por lo que numerosos tipos de máquinas son viables. Sin embargo, para la parte de la comparación frente a las bases de datos, las únicas máquinas viables de las estudiadas en este trabajo, son la m1.xlarge y la m3.large de la nueva generación. Las máquinas de la nueva generación superan con creces a las antiguas por lo que el siguiente esfuerzo sería analizar diferentes máquinas con estas características.

La rapidez del sistema se ve afectada tanto por el tamaño de la base de datos que utilizamos como por el tiempo mínimo de ejecución de una foto en cada una de las máquinas. Además, se ha probado que distintas distribuciones de trabajo en las máquinas con más de una CPU pueden mejorar el rendimiento del sistema. Conociendo el sistema de pay-per-use de AWS fraccionado por horas, se han presentado el número de máquinas óptimo para hacer frente a un determinado flujo de individuos en una hora, minimizando de esta manera los costes.

Utilizando el sistema para escenarios estáticos, minimizamos la interacción de las variables del entorno en la fiabilidad de las comparaciones, evitando la baja fiabilidad de otros sistemas de seguridad con reconocimiento facial que habían sido utilizados en el pasado. Sin embargo, la mejora en la tecnología del reconocimiento

facial permite que la fiabilidad en entornos dinámicos haya mejorado, hasta tal punto, que poseen una efectividad cercana a la efectividad de los entornos dinámicos, controlando parcialmente las variables del entorno.

La utilización conjunta de estos dos sistemas, suponen una gran mejora en la seguridad de cualquier lugar público, como se puede ver en el caso de uso presentado del aeropuerto de Barajas. Hoy en día, la seguridad es cada vez más importante por lo que este tipo de sistemas son cada vez más utilizados.

La principal contribución de este trabajo es la propuesta de dos arquitecturas para un sistema de reconocimiento facial cloud, el SRFC para escenarios estáticos y el SRFC para escenarios dinámicos, con el fin de solventar los problemas de fiabilidad y falta de rendimiento de estos sistemas. Estas arquitecturas definen, que tipo de máquinas hay que utilizar para hacer frente a una determinada carga de trabajo y a una base de datos de un tamaño concreto. El SRFCE permite utilizar el reconocimiento facial con una gran fiabilidad con una gran potencia de computación para hacer frente a cantidades muy grandes de individuos. El SRFCD permite captar más información en menos tiempo permitiéndonos utilizar este sistema como complemente del anterior.

Las arquitecturas propuestas son reflejadas en un árbol de decisión de infraestructura del sistema de seguridad con reconocimiento facial cloud, donde el usuario que necesite desplegar este sistema puede elegir fácilmente la mejor opción dependiendo de sus preferencias en términos de seguridad, coste y rendimiento. Este modelo determina qué máquinas de AWS son las más adecuadas para cada usuario.

Ahora, todos los esfuerzos ser pueden concentrar en mejorar tanto el rendimiento y los costes de este sistema para su evolución y aumento de sus características.

Capítulo 10

10. Trabajo Futuro

Las arquitecturas propuestas para un sistema de seguridad con reconocimiento facial en el cloud público resuelven algunos de los principales problemas de estos sistemas. La fiabilidad, la falta de poder de computación o los altos costes de su despliegue son algunos de los temas tratados en este trabajo. Cada una de las arquitecturas propuestas resuelve algunos de estos problemas pero no del todo, siendo esencial para la seguridad y para la mejora de estos sistemas investigaciones futuras en estos aspectos. El trabajo futuro sobre este tipo de sistemas incluye:

- **Mejora del rendimiento del sistemas a través del estudio de instancias más potentes de nueva generación.** En este trabajo se ha demostrado el gran rendimiento de estas máquinas por lo que existen numerosas opciones más potentes que beneficiarían a este tipo de sistemas. Dentro de este gran grupo, existen instancias con GPU, las cuales podrían ser utilizadas como aceleradores y mejorar notablemente el rendimiento del sistema gracias a su gran potencia de paralelización. Todas estas mejoras permitirían al sistema acercarse a la ejecución en tiempo real.

- **Mejora en las capacidades del sistema.** Por ejemplo, enlazar la información de las fotografías con información almacenada no solo en bases de datos de criminales, si no con redes sociales, bases de datos de organismos públicos o internet en general. De esta forma, estaríamos generando big data a través de la big data.

- **Ampliación de los escenarios de uso.** Con la suficiente potencia de computación y diversas mejoras del rendimiento, el sistema podría ser utilizado a pie de calle por los diversos cuerpos de seguridad. Su utilización junto con al nuevo producto de google, las google glass, podría ser una excelente herramienta para la policía en sus tareas de vigilancia a pie de calle.

- **Mejora en el cálculo de costes a través de redes neuronales.** La utilización de redes neuronales es muy útil a la hora de pronosticar el precio de un determinado recurso. Su utilización junto con la bolsa de mercado de AWS podría producir un gran abaratamiento del sistema como se puede apreciar en [59]. Además, su utilización junto a otros sistemas, como dinámicas de fluidos podrían predecir con gran exactitud el flujo de individuos y la cantidad de recursos necesarios para afrontarlos.

- **Estudio de otros proveedores cloud, incluso a otros niveles.** Existen una gran variedad de proveedores cloud con gran diversidad de instancias en cada una. Un estudio a gran escala de todos estos proveedores proporcionaría una mejor estrategia a la hora de invertir recursos económicos y mejorar el rendimiento del sistema. Además existe la posibilidad de ampliar el sistema a otros niveles de cloud, por ejemplo, Google App Engine está a nivel de PaaS y el código de nuestro sistema de reconocimiento facial podría ser ejecutado en su entorno.

Referencias

[1] A. K. Jain, R. Bolle y S. Pantanti, «Biometrics: Personal Identification in Networked Security,» *Kluwer Academic Publishers,* 1999.

[2] C. Nastar y M. Mitschke, «Real time face recognition using feature combination,» *Thrid IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 312-317, Nara, Japan, 1998.

[3] T. Jebara, «3D Pose Estimation and Normalization for Face Recognition,» *Center for Intelligent Machines, McGill University, Undergraduate Thesis,* May, 1996.

[4] R. Brunelli y T. Poggio, «Face recognition: features versus templates,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, nº 1042-1052, 1993.

[5] M. A. Grudin, «On internal representations in face recognition systems,» *Patter Recognition,* vol. 33, pp. 1161-1177, 2000.

[6] B. Heisele, P. Ho, J. Wu y T. Poggio, «Face Recognition: component-based versus global approaches,» *Computer Vision and Image Understanding,* vol. 91, pp. 6-21, 2003.

[7] T. Kanade, «Picture Processing System by Computer Complex and Recognition of Human Face,» *Kyoto University, Japan,* 1973.

[8] I. J. Cox, J. Ghosn y P. N. Yianilos, «Feature-based face recognition using mixture-distance,» *Proceedings of IEE Conference on Computer Vision and Pattern Recognition,* pp. 209-216, 1996.

[9] L. Wiskott, j. -M. Fellous, N. Krüger y C. von der Malsburg, «Face Recognition by Elastic Bunch Graph Matching,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 775-779, 1997.

[10] R. Cendrillon y B. C. Lowell, «Real-Time Face Recognition using Eigenface,» *Proceedings of the SPIE International Conference on Visual Communications and Image Processing,* vol. 4067 , pp. 269-276, 2000.

[11] R. J. Baron, «Mechanisms of Human Facial Recognition,» *International Journal of Man-Manchine Studies,* vol. 15, pp. 137-178, 1981.

[12] R. -J. J. Huang, «Detection Strategies for face recognition using learning and evolution,» *George Mason University, Fairfax, Virginia,* 1998.

[13] L. Sirovich y M. Kirby, «Low-dimensional Procedure for the Characterization of Human Face,» *Journal of the Optical Society of America A: Optics, Image Science, and Vision,* vol. 4, pp. 519-524, 1987.

[14] A. K. Jain y R. C. Dubes, «Algorithms for Clustering Data,» *New Jersey: Prentice-Hall,* 1988.

[15] K. Fukunaga, «Introduction to Statistical Patter Recognition,» *second ed. Boston, MA: Academic Press,* 1990.

[16] P. Comon, «Independent component analysis--A new concept?,» *Signal Processing,* vol. 36, pp. 287-314, 1994.

[17] R. A. Fisher, «The use of multiple measure in taxonomic problems,» *Annals of Eugenics,* vol. 7, pp. 179-188, 1936.

[18] J. Li, S. Zhou y C. Shekhar, «A Comparison of Subspace Analysis for Face Recognition,» *in Proc. IEEE Int'l Conf. on Acoustic, Speech, and Signal Processing. 2003,* pp. 121-124, 2003.

[19] Q. Yang y X. Tang, «Recent Advances in Subspace Analysis for Face Recognition,» *SINOBIOMETRICS,* pp. 275-287, 2004.

[20] S. Lawrence, C. L. Giles, A. C. Tsoi y A. D. Back, «Face Recognition: A Convolutional Neural Network Approach,» *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Patter Recognition,* pp. 1-24, 1997.

[21] U. Krebel, «Pairwise classification and support vector machines,» *Advance in Kernel Methods -- Suport Vector Learning,* pp. 255-268, 1999.

[22] C. Lui y H. Wechsler, «Evolutionary Pursuit and Its Application to Face Recognition,» *IEEE Transactions on Pattern Analysis and Machine Intelligence ,* vol. 22, pp. 570-582, 2000.

[23] H. -L. Huang, H. -M. Chen, S. -J. Ho y S. -Y. Ho, «Advanced Evolutionary Pursuit for Face Recognition,» *VLSI Signal Processing-System for Signal, Image, and Video Technology,* 2006.

[24] J. Lu y K. N. Plataniotis, «Boosting face recognition on a large-scale database,» *Proceedings of IEEE International Conference on Image Processing,* vol. 2, pp. 109-112, 2002.

[25] L. Torres, L. Lorente y J. Vilà, «Face recognition using self-eigenface,» *in International Symposium on Image/Video Communications Over Fixed and Mobile Networks. Rabat, Moroco,* pp. 44-47, 2000.

[26] R. Chellapa, C. L. Wilson y S. Sirohey, «Human and machine recognition of face: A survey,» *Proceedings of the IEEE,* vol. 83, pp. 705-740, 1995.

[27] A. Howell y H. Buxton, «Towards unconstrained face recognition from image sequences,» *in Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition, 1996,* pp. 224-229, 1996.

[28] T. E. de Campos, R. S. Feris y R. M. Cesar Jr., «A Framework for Face Recognition from Video Sequences Using GWN and Eigenfeature Selection,» *in Workshop on Artificial Intelligence and Computer Vision. Atibaia, Brazil,* 2000.

[29] V. Kruger y G. Sommer, «Affine real-time face tracking using a wavelet network,» *in ICCV'99 Workshop: Recognition, Analysis, and Tracking of Face and Gestures in Real-Time Systems.,* pp. 141-148, 1999.

[30] V. Krueger y S. Zhou, «Exemplar-based face recognition from video,» *in Computer Vision - ECCV 2002: 7th European Conference on Computer Vision,* vol. 2353, p. 732, May 28-31 2002.

[31] S. Zhou, V. Krueger y R. Chellapa, «Face Recognition from video: A CONDENSATION Approach,» *in Proc. of Fifth IEEE International Conference on Automatic Face and Gesture Recognition. Washington D.C,* pp. 221-228, 2002.

[32] J. Steffens , E. Elagin y H. Neven, «Person Spottter -- fast and robust system for human detection, tracking and recognition,» *in Proceedings, International Conference on Audio- and Video-Based Person Authentication,* pp. 96-101, 1999.

[33] T. Choudhry, B. Clarkson, T. Jebara y A. Pentland, «Multimodal person recognition using unconstrained audio and video,» *in Procceedings, International Conference on Audio and Video-Based Person Authentication,* pp. 176-181, 1999.

[34] L. Torres, «Is there any hope for face recognition?,» *in Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004). Lisboa, Portugal,* 2004.

[35] G. J. Liposcak y K. J. Breeding, «A scale-space approach to face recognition from profiles,» *in Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns,* vol. 1689, pp. 243-250, 1999.

[36] A. Tibbalds, «Three Dimensional Human Face Acquisition for Recognition,» *Trinity College, University of Cambridge, Cambridge, Ph. D. Thesis,* March 1998.

[37] J. -G. Wang , K. -A. Toh y R. Venkateswarlu, «Fusion of Appearance and Depth Information for Face Recognition,» *in Fifth International Conference on Audio- and Video-Based Biometric Person Authentication,* vol. 3546, pp. 919-928, 2005.

[38] C. Beumier y M. Acheroy , «Face verication from 3D and grey level clues,» *Pattern Recognition Letters,* vol. 22, pp. 4654-1329, 5114.

[39] P. Besl y N. McKay, «A method for registration of 3D shapes,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, pp. 239-256, 1992.

[40] C. Benabdelkader y P. Griffin, «Comparing and combining depth and texture cues for face recogniton,» *Image And Vision Computing,* vol. 23, pp. 339-352, 2005.

[41] R. Cutler, «Face recognition using infrared images and eigenfaces,» *University of Maryland at College Park, College Park, MD, USA,* 1996.

[42] X. Chen, P. Flynn y K. Bowyer, «Visible-light and infrared face recognition,» *Proceedings of the Workshop on Multimodal User Authentication. Santa Barbara, California, USA,* pp. 48-55, 2003.

[43] S. Rogerson, «Smart CCTV,» *ETHIcol in the IMIS Journal,* vol. 12, nº 1, February 2002.

[44] T. C. Greene, «http://www.theregiste.co.uk,» The Register, 27 Sep 2001. [En línea]. Available: http://www.theregister.co.uk/2001/09/27/face_recognition_useless_for_crowd/.

[45] J. Meek, «www.theguardian.com,» 13 June 2002. [En línea]. Available: http://www.theguardian.com/uk/2002/jun/13/ukcrime.jamesmeek.

[46] D. McCullagh, «Call It Super Bowl Face Scan I,» 2 February 2001. [En línea]. Available: http://archive.wired.com/politics/law/news/2001/02/41571.

[47] L. Greene, «Face scans match few suspects,» 16 February 2001. [En línea]. Available: http://www.sptimes.com/News/021601/TampaBay/Face_scans_match_few_.shtml.

[48] R. Trigaux, «Cameras scanned fans for criminals,» 31 January 2001. [En línea]. Available: http://www.sptimes.com/News/013101/TampaBay/Cameras_scanned_fans_.shtml.

[49] J. C. Klontz y A. K. Jain, «A Case Study of Automated Face Recognition: The Boston Marathon Bombings Suspects,» *ComputingNow,* November 2013.

[50] NEC, «NeoFace Solutions,» [En línea]. Available: http://www.necam.com/Biometrics/doc.cfm?t=FaceRecognition.

[51] «International Data Corporation,» April 2014. [En línea]. Available: http://www.emc.com/leadership/digital-universe/2014iview/index.htm.

[52] L. An, M. Kafai, Y. Songfan y B. Bhanu, «Reference-based person re-identification,» *Advanced Video and Signal Based Surveillance (AVSS), 10th IEEE International Conference,* pp. 244-249, 2013.

[53] P. Mell y T. Grance, «The NIST Definition of Cloud Computing,» *National Institute of Standars and Technology,* September 2011.

[54] C. Consulting, «Challenges & opportunities for it partners when transforming or creating a business in the cloud,» p. 77, 20012.

[55] Luxand, «www.luxand.com, Documentation,» 20013. [En línea]. Available: https://www.luxand.com/download/Luxand_FaceSDK_Documentation.pdf.

[56] «StarCluster,» [En línea]. Available: http://star.mit.edu/cluster/docs/latest/overview.html.

[57] Aena Aeropuertos, «Aena,» 2013. [En línea]. Available: http://www.aena-aeropuertos.es/csee/ccurl/113/554/estadisticas_anual_2013_provisionales.pdf. [Último acceso: 15 Jun 2014].

[58] Interpol, «Interpol,» [En línea]. Available: http://www.interpol.int/Maps. [Último acceso: 16 Jun 2014].

[59] R. M. Wallace , V. Turchenko, M. Sheikhalishahi, I. Turchenko, V. Shults , J. Vázquez-Poletti y L. Grandinetti, «Applications of Neural-based Sport Market Prediction for Cloud Computing,» *7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAAC S'2013) Berlin (Germany),* vol. 2, pp. 710-716, September 2013.